ARL LIBRARY (APG)
5 0592 01024294 4

PROPERTY OF U.S. ARMY

ARBRL-TR-02060
ADA056564

REFERENCE [redacted] ULATE

AD A056564

TECHNICAL REPORT ARBRL-TR-02060

A COMPUTER MAN ANATOMICAL MODEL

Charles A. Stanley
Michael Brown

PROPERTY OF U.S. ARMY
STINFO BRANCH
BRL, APG, MD. 21005
NOV 1980
COUNTED IN

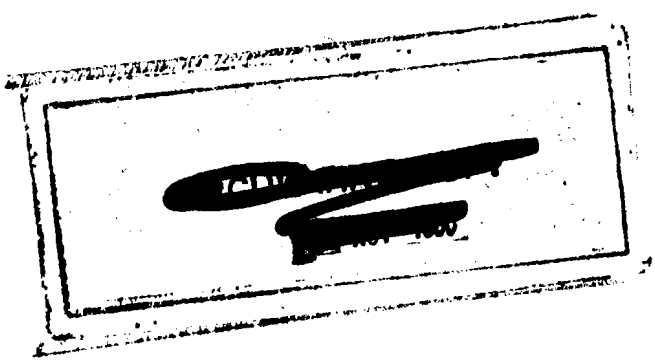
May 1978

REFERENCE
DOES NOT CIRCULATE

US ARMY ARMAMENT RESEARCH AND DEVELOPMENT COMMAND
BALLISTIC RESEARCH LABORATORY
ABERDEEN PROVING GROUND, MARYLAND

Approved for public release; distribution unlimited.

TECHNICAL LIBRARY
ARL [illegible] LABORATORY
ABERDEEN PROVING GROUND



Destroy this report when it is no longer needed.
Do not return it to the originator.

Secondary distribution of this report by originating or sponsoring activity is prohibited.

Additional copies of this report may be obtained from the National Technical Information Service, U.S. Department of Commerce, Springfield, Virginia 22161.

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

The use of trade names or manufacturers' names in this report does not constitute indorsement of any commercial product.

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: TECHNICAL REPORT ARBRL-TR-02060
2. GOVT ACCESSION NO.:
3. RECIPIENT'S CATALOG NUMBER:

4. TITLE (and Subtitle): A COMPUTER MAN ANATOMICAL MODEL
5. TYPE OF REPORT & PERIOD COVERED: Final
6. PERFORMING ORG. REPORT NUMBER:

7. AUTHOR(s): Charles A. Stanley, Michael Brown
8. CONTRACT OR GRANT NUMBER(s):

9. PERFORMING ORGANIZATION NAME AND ADDRESS: US Army Ballistic Research Laboratory, ATTN: DRDAR-BLV, Aberdeen Proving Ground, MD 21005
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 1L662628AH80

11. CONTROLLING OFFICE NAME AND ADDRESS: US Army Armament Research & Development Command, US Army Ballistic Research Laboratory (DRDAR-BL), Aberdeen Proving Ground, MD 21005
12. REPORT DATE: MAY 1978
13. NUMBER OF PAGES: 265

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office):
15. SECURITY CLASS. (of this report): UNCLASSIFIED
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

PROPERTY OF U.S. ARMY
STINFO BRANCH
BRL, APG, MD. 21005

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Lethality
Incapacitation
Medical Assessment
Grid Cell
Wounding Process
Personnel Vulnerability

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Computer Man Anatomical Model report describes the arrangement of human body tissue in three-dimensional space. The Computer Man simulates a combat soldier engaged in military operations and functions as a model (target) for personnel vulnerability analyses.

This report gives a detailed comprehensive description of the Computer Man model. Complete documentation of all the required programs is provided.

TABLE OF CONTENTS

		Page
	LIST OF ILLUSTRATIONS	5
I.	INTRODUCTION	7
II.	VULNERABILITY	7
III.	COMPUTER MAN TARGET DESCRIPTION	9
	A. Origin of Computer Man	9
	B. Description Used for Incapacitation Studies	9
	C. Description Used for Lethality Studies	9
	D. Assembly of Cross Sections	9
	E. Positioning the Computer Man in Three-Dimensional Space	13
	F. Computer Man Body Box	13
	G. Cross-Sectional Oddities	17
IV.	COMPUTER MAN FILES	19
	A. Card Description	19
	B. Tape Description	20
	C. Disc Description	26
V.	COMPUTER MAN PROGRAMS - LETHALITY	28
	A. Disc Load and Average Score Programs	28
	B. Change Value Program	33
	C. Tape Translation Program	35
	D. Computer Man Delineation Program	39
	E. Instructions for Using Lethality Programs	41
VI.	COMPUTER MAN PROGRAMS - INCAPACITATION	42
	A. Model Repositioning	42

TABLE OF CONTENTS (Continued)

		Page
VI.	COMPUTER MAN PROGRAMS - INCAPACITATION (continued)	42
	B. Foot Slice Addition.	44
	C. Mosaic Printout.	46
	APPENDIX A. Assignment of Tissue Codes.	49
	APPENDIX B. Slice Descriptions - Incapacitation	59
	APPENDIX C. Slice Descriptions - Lethality.	145
	APPENDIX D. Flowcharts and Programs	231
	DISTRIBUTION LIST.	263

LIST OF FIGURES

Figure		Page
1.	A Front View of the Computer Man Indicating the Approximate Intervals at Which Slices Were Taken in the Thorax and Abdomen	10
2.	A Cross Section Taken from "A Cross-Section Anatomy" overlayed with Grid Mesh.	11
3.	A Pictorial View of a Chosen Cross Section Illustrating the Positioning and Contents of the Rectangular Blocks.	12
4.	A Visual Representation (Front View) of a Computer Man Description	14
5.	A Visual Representation (Side View) of a Computer Man Description	15
6.	The Positioning of the Computer Man in a Right-Hand Coordinate System.	16
7.	An Illustration Depicting the Ordering Procedure Employed in Generating Card Files	21
8.	Card Format of Cross Sections Representing the Initial Three Cuts Taken in the Head.	22
9.	An Illustration of the Positioning of the Card Data Inside the Target Matrix Using the Specified Parameters.	23
10.	Card File Configuration	24
11.	A Representation of the Disc Structure of the Computer Man (Lethality)	29
12.	A Representation of the Disc Structure of the Computer Man (Incapacitation).	45

I. INTRODUCTION

The following is a report on the design and development of the Computer Man for application to vulnerability studies of military personnel engaged in combat environments and subjected to various weapon threats. Although this model is not currently in use at the Ballistic Research Laboratory (BRL), it is presented in order to document the work done, and because the model may be of interest in future personnel vulnerability studies.

For many years injury criteria were based upon medical assessments of autopsy reports derived from animal experiments. However, attempts to correlate the effects of physiological damage in humans with that produced in animals led to certain difficulties. To avoid these problems, it was decided in 1974 to shift the personnel vulnerability effort to that of modeling the wounding process. Hence, the development of the Computer Man came into existence. The Computer Man Anatomical Model is part of an overall effort that has been conducted in the Personnel Vulnerability Group to develop an analytical model which could be used to predict the effects wounds have on human functions for various classes of projectiles.

The computer generated three-dimensional representation of a human male is defined as the Computer Man. The Computer Man is essentially an organized array of tissue codes which are encountered when theoretically generated random trajectories strike and penetrate or perforate the body.

The essence of this report consists of instructions and programs written to construct, store and print Computer Man models. Detailed documentation is provided for their interpretation and use.

II. VULNERABILITY

In the vulnerability analyses, the development of the criteria relating to the effects wounds have on a soldier's ability to perform in a combat environment, and the scope of the methodology used, is quite extensive.[1] Moreover, since incapacitation and lethality are the basic injury classifications studied in formulating the codes which make up the Computer Man, these two areas of interest will be discussed briefly here.

For the incapacitation data used with the Computer Man Anatomical Model described in this report, surgeons were asked to evaluate a list

---

1. *W. Kokinakis and J. Sperrazza, "Criteria for Incapacitating Soldiers with Fragments and Flechettes," Ballistic Research Laboratories Report No. 1269, Jan 1965.* ([redacted]) (AD #359774)

of 181 tissues which, when rendered non-functional by a penetrating projectile, could induce some level of incapacitation with respect to a given scenario. A detailed description of the tissue codes which define the Computer Man is listed for reference in Appendix A. Three time frames were stipulated in conjunction with each of two classifications of combat roles:

1. assault immediate.
   defense immediate.
2. assault $\leq$ 30 sec.
   defense $\leq$ 30 sec.
3. assault $\leq$ 5 min.
   defense $\leq$ 5 min.

Surgeons used a range of whole numbers (0-100) to indicate quantitatively the level of incapacitation rendered. Pain was of no consideration. Approximately a third of the anatomical tissues were assigned non-zero scores.

To generate the tools for the lethality analysis a different approach was undertaken. A team of surgeons was presented a set of gridded anatomical cross sections as developed by Eycleshymer and Schoemaker.[2] The medical assessors were asked to independently estimate the effects of the removal of a defined quantity of body tissue on human survivability. A relative scale of whole numbers (1-10) was employed. A score was entered for each tissue cell defined on a cross section based upon the probability of death occurring, given the removal of a single cell. Scenarios were established for four time frames relative to the availability of expert medical aid; within 30 minutes, within one hour, within six hours, and no treatment.

The number of assessments differ for each of the preceding time frames. Initially, three assessors were asked to make estimates on all cells throughout the body. Subsequently, an additional group of assessors were asked to provide estimates for those parts of the body for which they were considered specialists. In these cases the body was divided into four sections; head and neck, thorax, abdomen and pelvis, and limbs. Composite evaluations are treated in the same manner as individual evaluations. Specific details on the medical judgment process for estimating incapacitation, as well as lethality, and the methodology used to correlate damage with incapacitation/lethality, will be documented in a BRL publication which is presently being organized. In the interim, queries concerning the medical inputs to the Computer Man Anatomical Model should be addressed to Director. Ballistic Research Laboratory, ATTN: DRDAR-BLV, Aberdeen Proving Ground, MD 21005.

2. *A.C. Eycleshymer and D.M. Schoemaker, "A Cross-Section Anatomy," 215pp, D. Appleton - Century Company, New York and London, 1911.*

III. COMPUTER MAN TARGET DESCRIPTION

A. Origin of Computer Man

Physical dimensions of the Computer Man are based upon photographs taken from the Cross-Section Anatomy of reference 2, in which a set of 108 horizontal cross sections pertaining to an adult male are given. The anatomical cross sections are grouped according to body parts; head and neck (cross sections 1-18); thorax, abdomen and pelvis (cross sections 19-44); left arm (cross sections 50-75); left leg and foot (cross sections 76-113). Cross sections (45-49) are descriptions of the female organs taken in the pelvic region and are not included in our study. Figure 1 illustrates the approximate intervals at which slices were taken from the human anatomy in the thorax, abdomen and pelvis. The spacing between slices is 2.6 cm.

The initial step in the Computer Man development is conducted by superimposing a grid mesh onto each photograph and examining the contents of each cell present. A representative cross section is illustrated in Figure 2. A set of tissue codes was devised based upon the principle tissue contained in each cell and each cross section was stored in computer memory as a two-dimensional array. A pictorial view of a typical cross section is displayed in Figure 3. The integers present in the cells for this illustration indicate lethality scores.

B. Description Used for Incapacitation Studies

For incapacitation the dimensions of each grid cell are .5cm x .5cm x 1.2cm in the head and neck; throughout the remainder of the body the dimensions are .5cm x .5cm x 2.6cm. The contents of each cell is a tissue code which is an integer in the range (2-200). The incapacitation estimates assigned by medical assessors are not part of the target description but can be cross referenced with their associated tissue codes whenever a computation is required.

C. Description Used for Lethality Studies

For lethality the dimensions of each cell are 1cm x 1cm x 1.2cm in the head and neck; throughout the rest of the body the dimensions are 1cm x 1cm x 2.6cm. The contents of each cell is an assessor's evaluation, an integer in the range (1-10).

D. Assembly of Cross Sections

All arrays are grouped into a superset consisting of 108 elements (slices). Each array consists of a number of rows and columns. The row and column width is dependent on the area of each cross section and the mesh of each grid used. By stacking and attaching each slice in its properly related position, the initial set of 108 anatomical cross sections given in reference 2 evolved into a three-dimensional model

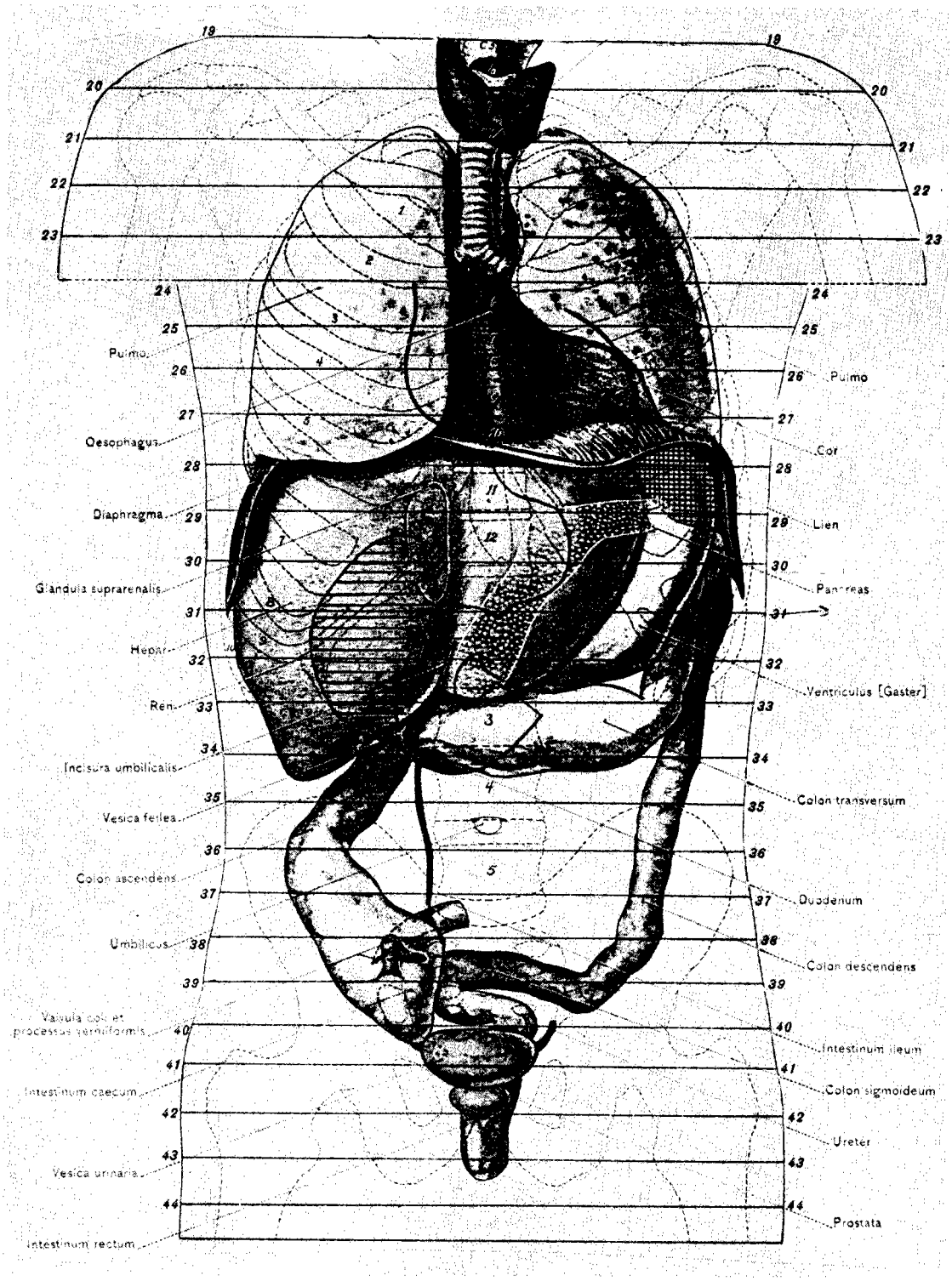


Figure 1. A Front View of the Computer Man Indicating the Approximate Intervals at which Slices Were Taken in the Thorax and Abdomen

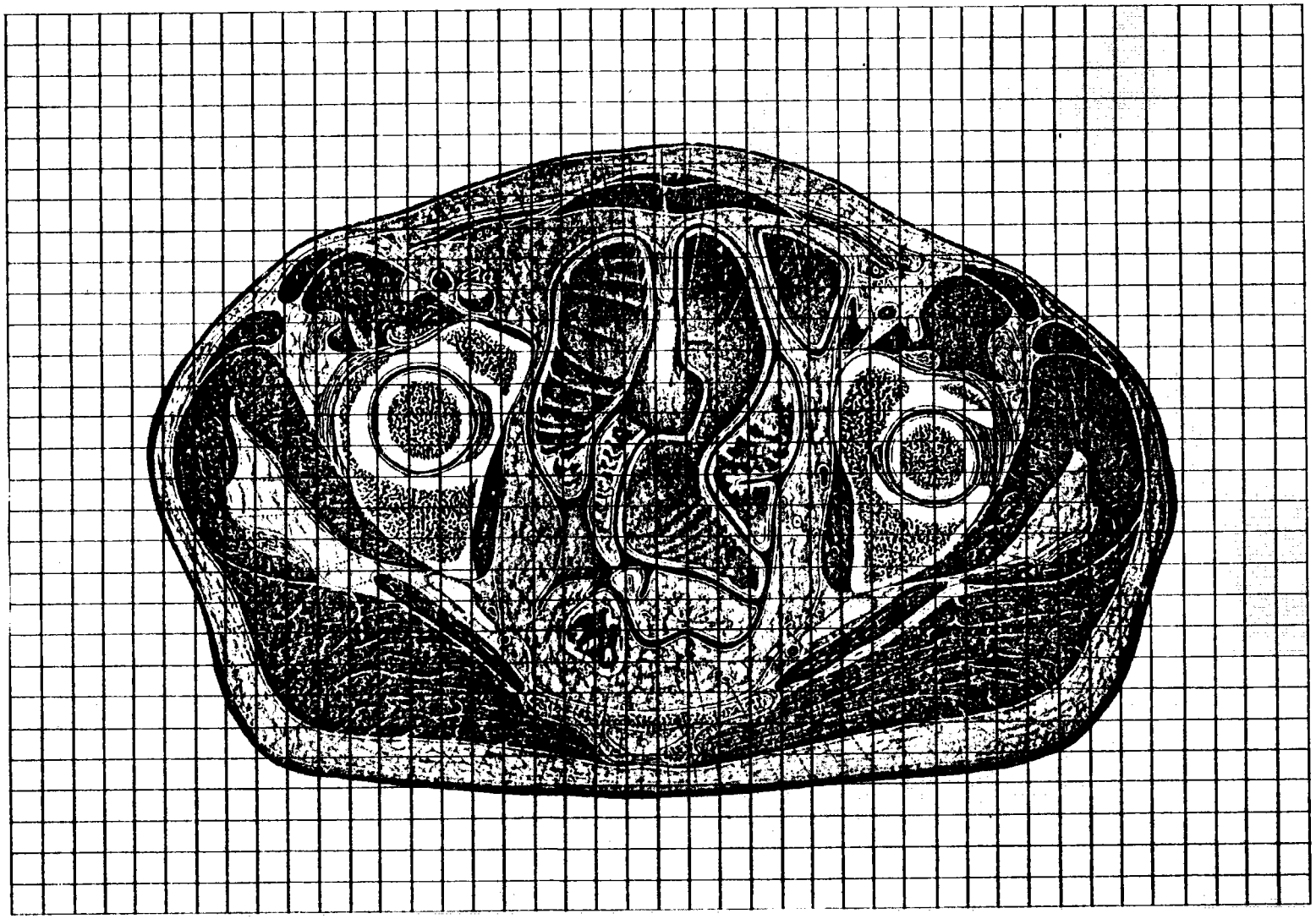


Figure 2. A Cross Section Taken From "A Cross-Section Anatomy" Overlayed with Grid Mesh

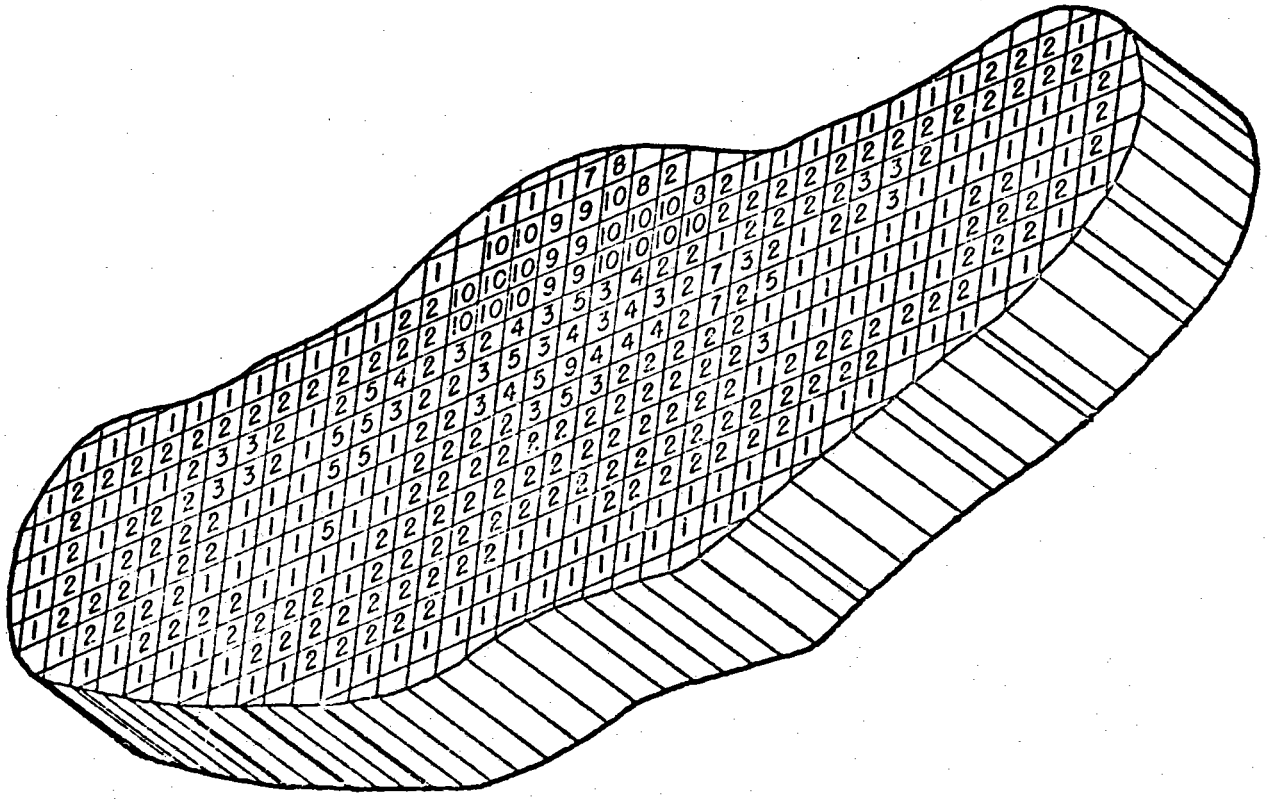


Figure 3. A Pictorial View of a Chosen Cross Section Illustrating the Positioning and Contents of the Rectangular Blocks

consisting of 82 composite cross sections. Figures 4 and 5 provide a visual characterization of the form the Computer Man assumes when stored in computer memory as a three-dimensional array. The right limbs were generated by merely reflecting the left limbs about the center line of the body.

In stacking the slices, a line of orientation was necessary to ensure that each cross section was aligned vertically with adjacent cross sections. This was accomplished through benchmarks between slices. In the arms and legs, large bones served as benchmarks. In the head, smaller slices were arranged above and below others by treating each slice as an elipsoid and matching their relative centers. In the torso, the vertebrae was employed as a benchmark. A more accurate procedure for establishing and utilizing benchmarks was not possible since cross-sectional anatomy data were prepared from a number of cadavers and body parts each of which differed in height, weight, age, and cross-sectional area.

E. Positioning the Computer Man in Three-Dimensional Space

The Computer Man can be pictured as standing in a right-hand coordinate system as portrayed in Figure 6-A. If the coordinate axes are rotated so that the Computer Man can be viewed in his prone position, as illustrated in Figure 6-B, the Computer Man model slice descriptions can be easily interpreted. If slices are cut parallel to the xy plane, starting at the origin, the first body cells encountered are cells comprising the feet. Thereby the Computer Man was assembled feet first progressing in the +z direction until the total configuration was complete.

Appendix B lists a slice by slice computer description of the incapacitation model derived by substituting a symbol for each tissue code. Appendix A lists the symbolic identification code, cross sections where tissues are located and physical structure associated with each tissue code defined in the model.

Appendix C contains a slice by slice computer description of the lethality model. The average of all doctors' evaluations for the one hour time frame is given.*

F. Computer Man Body Box

The target analysis, specifically the ray tracing routines, was simplified by enclosing the Computer Man inside a rectangular box. The box is called the body box and its dimensions are slightly larger than those of the Computer Man. The boundaries of the body box are defined by the size of the three-dimensional array, MAN(I,J,K). It is

---

**Slice descriptions representing additional time frames can be procured through the Target Assessment Branch, Vulnerability/Lethality Division, BRL.*

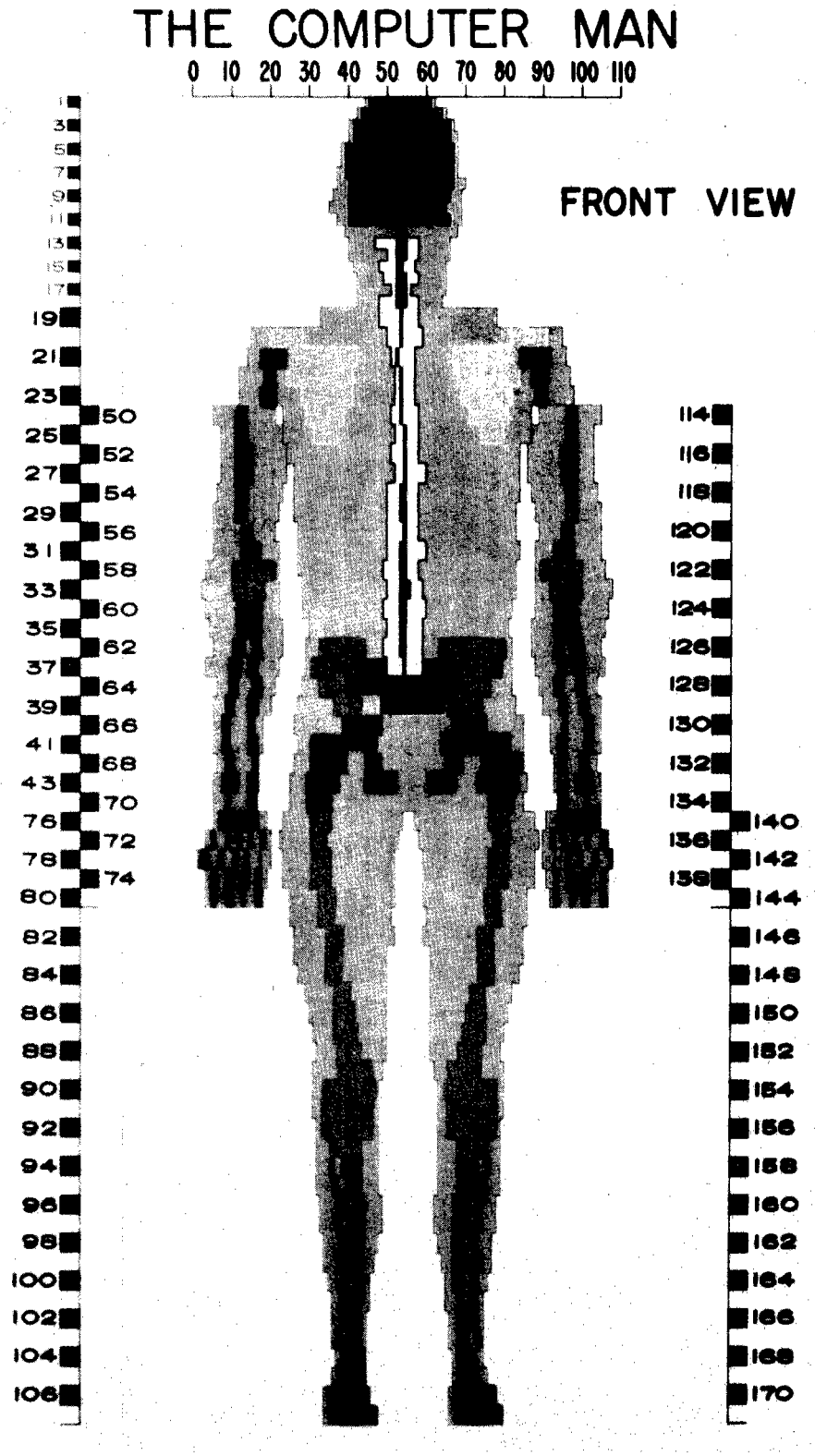


Figure 4. A Visual Representation of a Computer Man Description

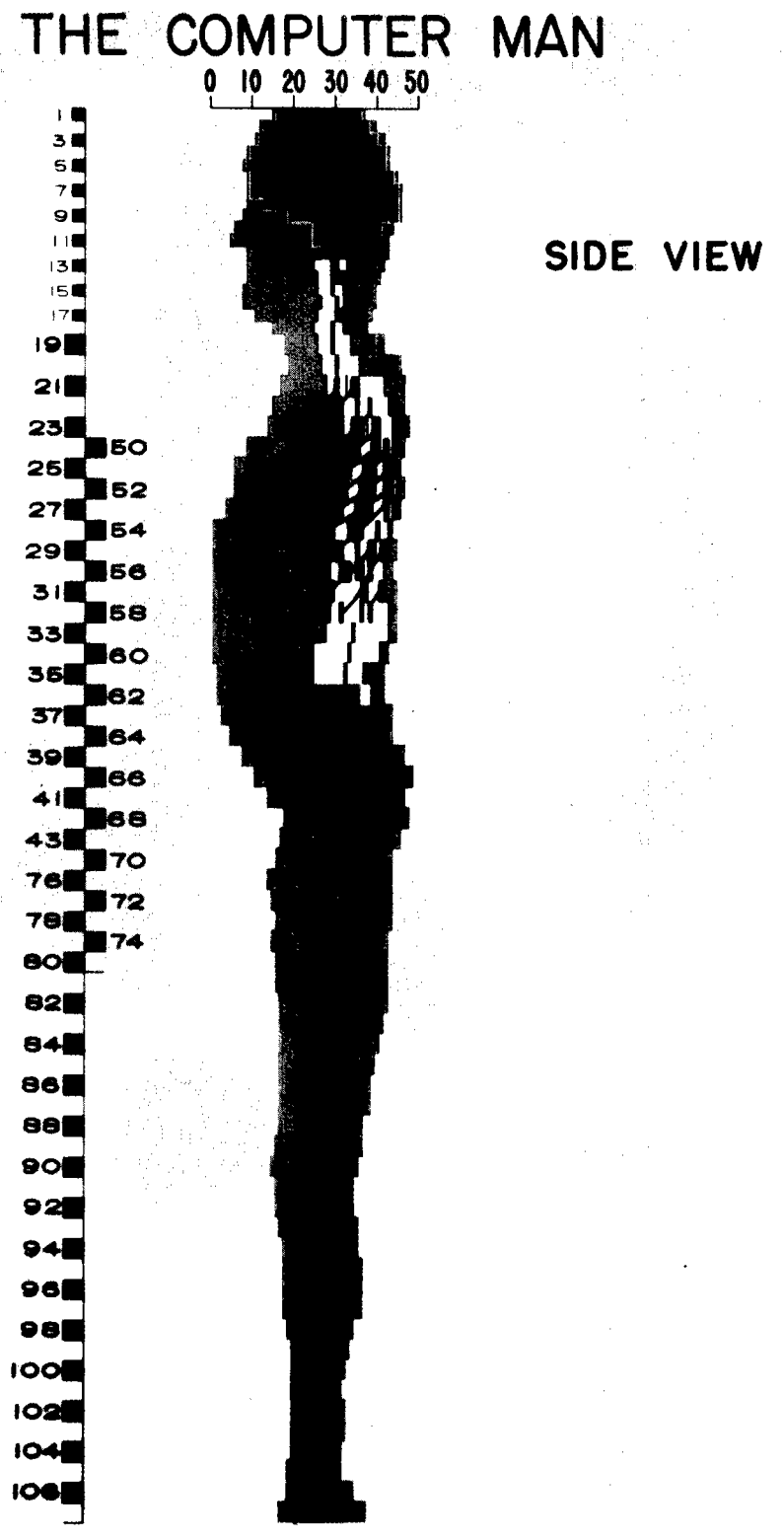


Figure 5. A Visual Representation of a Computer Man Description

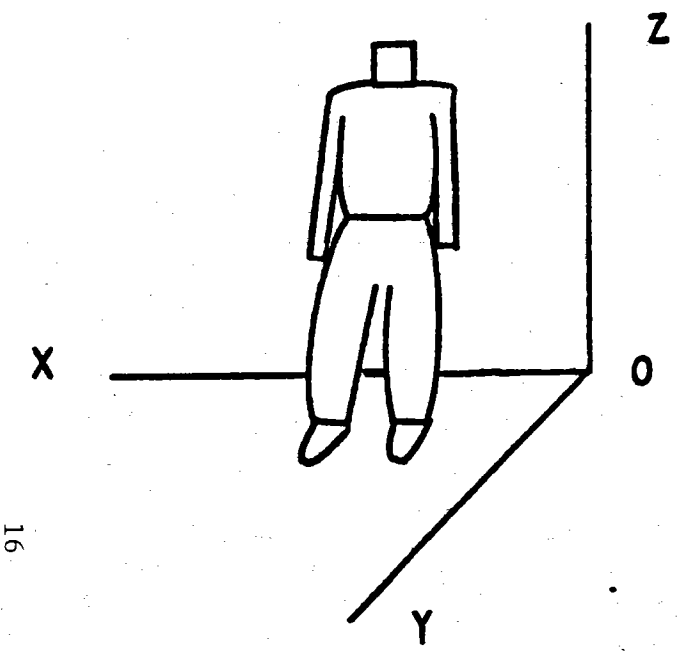


Figure 6-A.

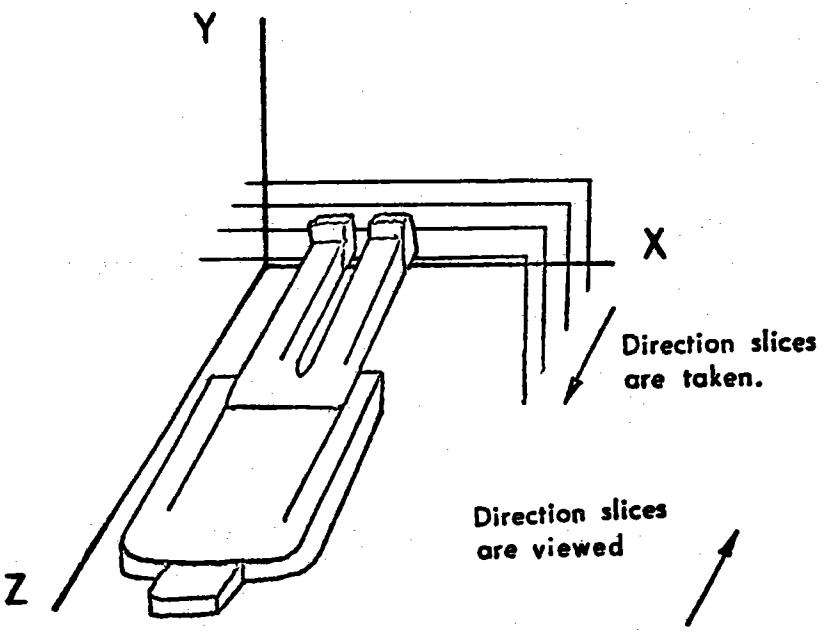


Figure 6-B.

Figures 6-A, 6-B. The Positioning of the Computer Man in a Right-Hand Coordinate System

positioned entirely within the first quadrant with the origin serving as one of its vertices. The cellular structure for each Computer Man cross section extends to the edges of the box. As a result, many cells are incorporated into the Computer Man model which represent air space. These cells are identified with zeros. Each cell within the body box can be referenced in computer memory by a unique set of I,J,K coordinates. Since ray tracing is conducted in real space it is necessary to have a real characterization of the model. The specific dimensions of the body box and body cells are listed for both models in Table I.

G. Cross-Sectional Oddities

1. Arms. In viewing arm slices of the Computer Man, some peculiarities are apparent in both the lethality and incapacitation models. It appears that the left and right arm slices are not equally spaced with respect to the torso. One arm will appear to be merged with the torso while the opposite arm will be separated from the torso by two or three units. The condition varies from slice to slice but is evident on most of the cross sections taken in this region. There are three explanations for this.

a. The shoulder slices do not fit into the grid system in a symmetrical fashion. Therefore the arms are misaligned relative to one another from the initial point where they are attached to the shoulders.

b. The arms are intentionally raised slightly upward and outward from the body to correct the condition where the hands would otherwise overlap the hips.

c. The third reason lies in the cells comprising the torso. The torso cross sections are not all of the same column and row width, thus after stacking every body cell on or adjacent to the perimeter on one slice does not necessarily have a body cell adjoined on the slice above or below it.

2. Torso. The outline of cells which form the perimeter of the Computer Man on each slice viewed may not always look symmetrical; that is, some cells or a number of consecutive cells may appear to be missing. Actually they are not. The reason for this is that the entire grid square is allocated even if only a small portion of body tissue would overlap the edge of this square, and thus when translated into its numeric or mosaic configuration, produces a jagged perimeter. This condition is primarily evident in the torso section.

Finally, as stated earlier, erratic differences in physical dimensions from slice to slice are inherent due to the fact, that cross-sections taken from the human anatomy were not all taken from the same man,but in fact represent a number of different individuals. The result is a Computer Man misaligned in some areas.

Table I. Graphical Dimensions of the Computer Man

DIMENSIONS OF BODY BOX

INCAPACITATION		LETHALITY	
(I,J,K) COORDINATES	REAL (X, Y, Z)	(I, J, K) COORDINATES	REAL (X, Y, Z)
(110, 55, 84)	(55 cm., 27.5 cm., 190.4 cm.)	(50, 35, 84)	(50cm., 35cm., 190.4 cm.)

DIMENSIONS OF BODY CELLS

	INCAPACITATION (X,Y,Z)	LETHALITY (X,Y,Z)
HEAD SLICES (1 — 18)	(.5cm., .5cm., 1.2 cm.)	(1cm., 1cm., 1.2cm.)
BODY SLICES (19—113)	(.5cm., .5cm., 2.6 cm.)	(1cm., 1cm., 2.6 cm.)

IV. COMPUTER MAN FILES

A. Card Description

The initial step in storing doctors' assessments was conducted by generating a card file. This was done by copying the doctors' assessments for each gridded cross section provided onto card coding sheets and in turn onto punched cards.

In the case of incapacitation, a card description of the Computer Man was not constructed because an accurate target description already existed. However, the scheme discussed here, defined for lethality, is applicable to both models.

A card file was generated by placing the arguments NUM and SIG on each card where:

(1) NUM = slice number given in Eycleshymer and Schoemaker.

(2) SIG = a code number denoting the type of card being read.

-1 - denotes a set of positioning parameters.

1 - denotes the first 23 entries to a given row follow.

2 - denotes the second 23 entries to a given row follow.

3 - denotes the third 23 entries to a given row follow.

4 - denotes the fourth 23 entries to a given row follow.

5 - denotes the fifth 23 entries to a given row follow.

The initial card (first card) which precedes each slice description contains six additional arguments MNR, MXR, MNC, MXC, NCOL, NROW, which assist in positioning the scores in the matrix A(I,J) that stores each slice description where:

(3) MNR = minimum row number in submatrix.

(4) MXR = maximum row number in submatrix.

(5) MNC = minimum column number in submatrix.

(6) MXC = maximum column number in submatrix.

(7) NCOL = number of columns in submatrix.

(8) NROW = number of rows in submatrix.

Immediately following the slice arguments are the scores associated with the grid cells that are present on each row defined on the cross-section. The cards which contain doctor assessments are called score cards. For a fixed grid mesh the number of score cards necessary to describe each individual slice is dependent on two factors: (1) the row and column width, and (2) the I/O format chosen.

For any given slice, score cards are arranged in the file rowwise in accordance with the following procedure. The card associated with the posterior and left most portion of the Computer Man was stacked immediately behind the first card (refer to Fig. 7). All the cards necessary to describe the back row follow consecutively until the right most portion of his body is completed. For lethality, two cards are sufficient to describe the row with the largest number of entries. The next row and each subsequent row that follows were stacked in the same manner until the last card, the card pertaining to the front row and right most portion of his body, was placed. The score cards consist of only the codes which pertain to body tissue. The remaining cells which are scored as zeros are omitted. Each succeeding slice description was then stacked in numerical order behind its predecessor until all slices were filed. The card format of cross sections representing the initial three cuts taken in the head is illustrated in Fig. 8. The matrix in Fig. 9 illustrates the positioning of a representative slice (slice 2) inside the array A(I,J) using the positioning parameters specified on the first card.

In building the Computer Man, slice positioning was difficult because each row of a given slice varied in width. So to simplify matters, the character "B" was chosen to fill in cells in the grid mesh which did not contain anatomical tissue, but could be used to square off a submatrix, thus giving each row of a given slice an equal number of cells and each score card an equal number of score positions. This technique facilitated positioning the submatrix inside A(I,J).

Four card files were constructed, one for each time frame stipulated. Each card file was divided into N sections where N is the number of separate doctors' assessments provided for each time frame. The sections can be organized in various ways but for assignment to disc storage they must be grouped as illustrated in Fig. 10.

B. Tape Description

Tabs were made of the card description and corrections were entered when errors were revealed. The inspection and correction process was continued until an accurate description was obtained. Finally, each card file was loaded onto magnetic tape for permanent storage.

By storing the target description on an auxiliary peripheral, such as a tape or disc, and reading selected portions into core, when needed, the usage of the Computer Man can be fully accommodated with reasonable efficiency.

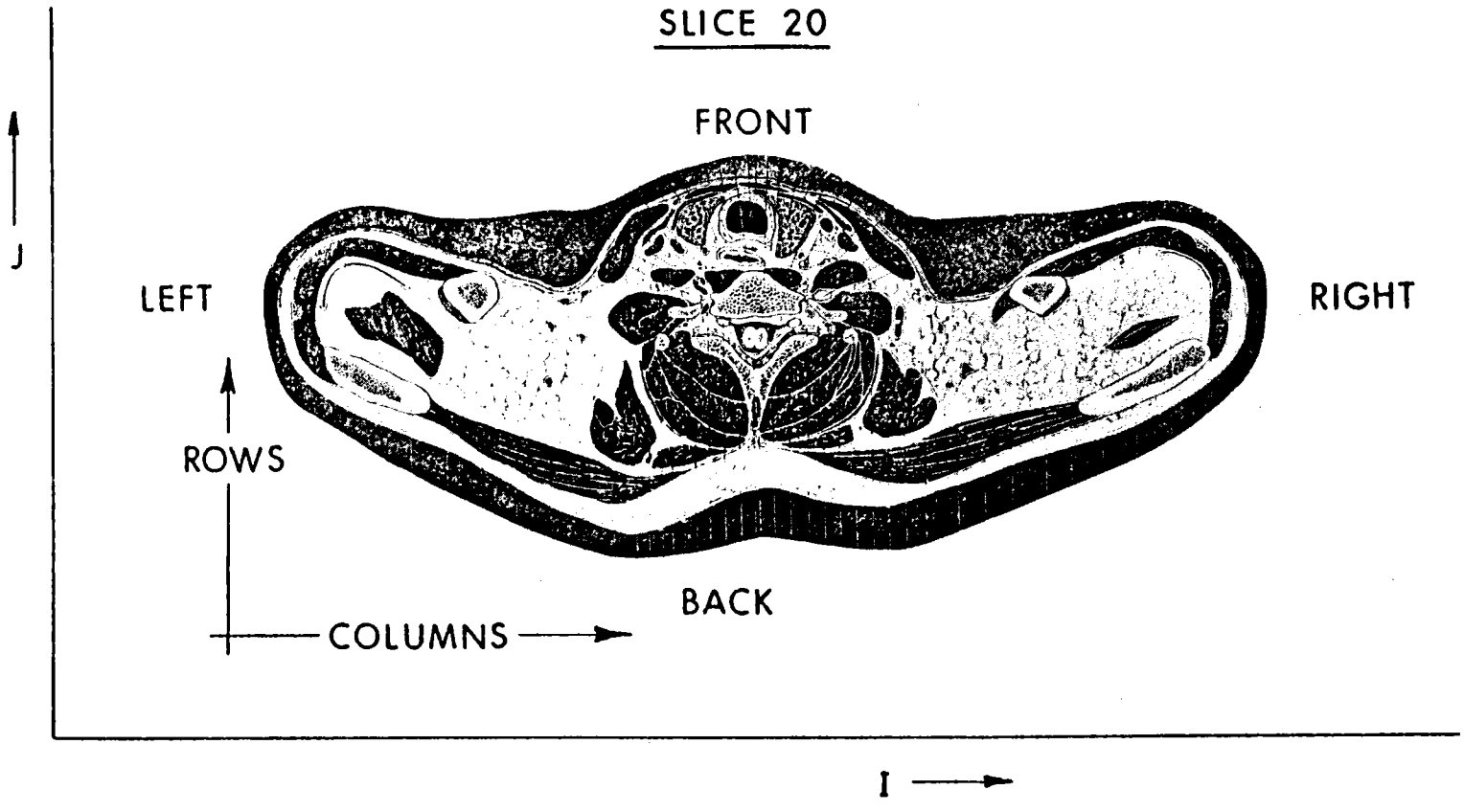


Figure 7. An Illustration Depicting the Ordering Procedure Employed in Generating Card Files

```
1-1   15   27   19   29        11   13
1 1   B    B    1    1    1    1    1    1    1    B    B
1 1   B    1    1    1    1    2    2    2    2    1    B
1 1   1    1    2    2    4    6    4    4    3    2    1
1 1   1    2    2    2    4    9    6    5    3    2    1
1 1   1    2    2    2    4    9    6    5    4    3    1
1 1   1    2    2    2    4    9    6    6    5    3    1
1 1   1    2    2    2    4    9    6    6    4    3    1
1 1   1    2    2    3    4    9    9    5    4    3    1
1 1   1    3    4    5    5    9    9    5    3    2    1
1 1   1    2    3    4    5    9    5    4    3    2    1
1 1   1    1    2    3    4    4    3    3    2    1    B
1 1   B    B    1    2    3    3    3    2    1    1    B
1 1   B    B    B    1    1    1    1    1    B    B    B
2-1   14   29   18   31        14   16
2 1   B    B    B    1    1    1    1    1    1    1    1    B    B    B
2 1   B    B    1    1    1    2    2    2    2    2    1    1    B    B
2 1   B    1    1    2    3    3    3    4    4    4    3    2    1    B
2 1   B    1    2    3    5    5    9    9    5    5    5    4    1    1
2 1   1    1    3    5    5    5    9    5    5    5    5    4    2    1
2 1   1    1    3    5    5    5    5    5    5    5    5    5    3    1
2 1   1    2    6    7    5    5    5    5    5    5    5    5    3    1
2 1   1    2    6    7    5    5    5    5    5    5    5    5    3    1
2 1   1    2    7    5    5    5    5    5    5    5    5   10    3    1
2 1   1    2    4    5    5    5    5    5    5    5    5    5    2    1
2 1   1    1    3    5    5    5    5    5    5    5    5    5    2    1
2 1   B    1    3    4    5    5    5    5    5    5    5    3    1    B
2 1   B    1    2    3    4    5    9    5    5    5    3    2    1    B
2 1   B    B    1    2    3    3    3    4    4    3    2    1    B    B
2 1   B    B    B    1    1    2    3    3    3    2    1    B    B    B
2 1   B    B    B    B    B    1    1    1    1    B    B    B    B    B
3-1   12   30   17   32        16   18
3 1   B    B    B    B    B    1    1    1    1    1    1    B    B    B    B    B
3 1   B    B    B    1    1    1    2    2    2    2    2    1    1    B    B    B
3 1   B    B    1    1    2    3    3    3    4    4    4    4    2    1    B    B
3 1   B    B    1    2    5    5    5    9    8    5    5    5    4    2    1    B
3 1   B    1    2    5    5    5    5    9    5    5    5    5    5    4    1    1
3 1   B    1    3    5    5    5    5    5    5    5    5    5    5    5    2    1
3 1   1    2    5    5    5    5    5    5    5    5    5    5    5    5    3    1
3 1   1    2    5    5    5    5    5    5    5    5    5    5    5    6    4    1
3 1   1    2    5    5    5    5    5    5    5    5    5    5    5    5    4    1
3 1   1    2    6    5    5    5    5    5    5    5    5    5    5    6    3    1
3 1   1    2    6    5    5    5    5    5    5    5    5    5    5    6    2    1
3 1   B    2    4    5    5    5    5    5    5    5    5    5    5    4    2    1
3 1   B    1    3    5    5    5    5    5    5    5    5    5    5    3    1    B
3 1   B    1    2    3    5    5    5    5    5    5    5    5    4    3    1    B
3 1   B    1    1    3    4    5    5    7    5    5    5    4    3    2    B    B
3 1   B    B    1    2    3    4    5    9    5    5    4    3    2    1    B    B
3 1   B    B    B    1    2    2    3    3    4    3    2    1    1    B    B    B
3 1   B    B    B    B    B    1    1    1    1    1    1    B    B    B    B    B
```

Figure 8. Card Format of Cross Sections Representing the Initial Three Cuts Taken in the Head

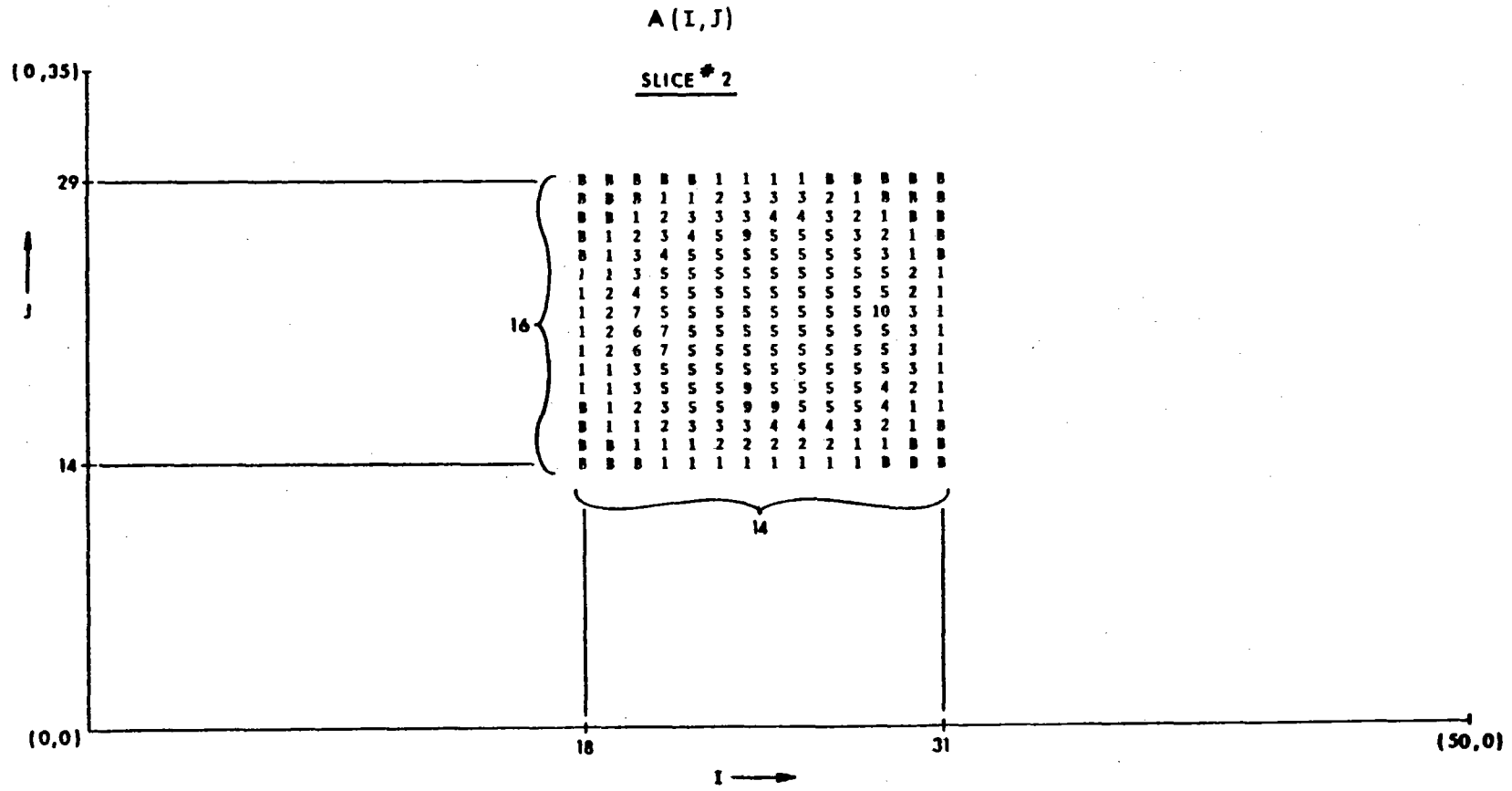


Figure 9. An Illustration of the Positioning of the Card Data Inside the Target Matrix Using the Specified Parameters

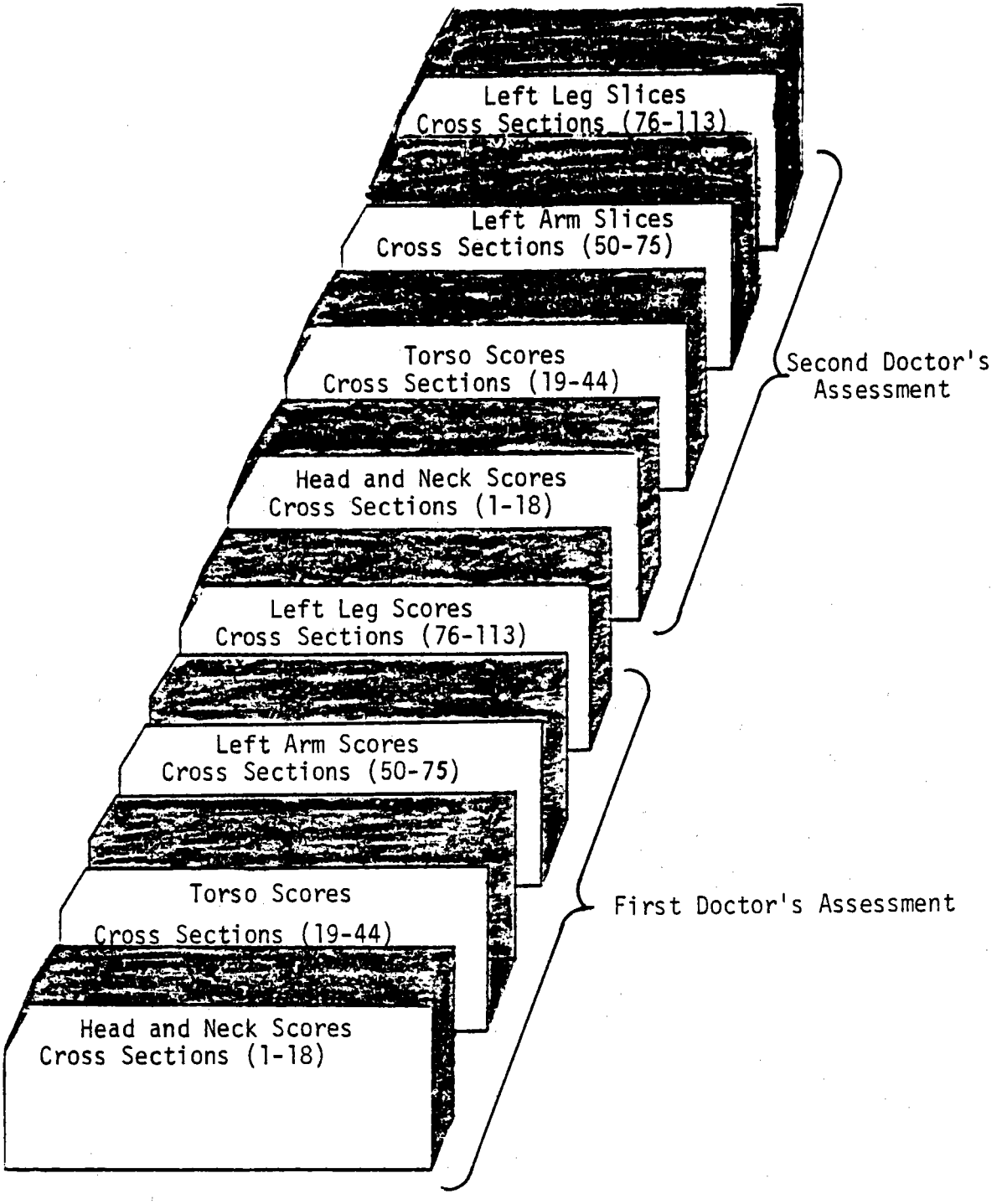


Figure 10. Card File Configuration

In early descriptions some doctors extended the range of scoring by using 0 as the lowest lethality score. This caused a problem because no distinction could be made between the minimum lethality score and code designated to represent air space. To resolve the problem each doctor's assessment was increased by 1, thus shifting the range of scoring from 0-10 to 1-11. Shortly afterwards the minimum lethality score was dropped. A tape translation program was written to facilitate these changes without having to punch new cards. In the revised edition the scores range in value (2-11).

The steps required for reading each cross section of the Computer Man into computer memory from tape are outlined below:

(1) find SIG.

(2) zero A(I,J).

(3) read in positioning parameters defined on first card.

(4) read in codes and position submatrix in A(I,J) designated by MNR, MXR, MNC, MXC.

Since slice descriptions were filed in the manner discussed in Section VI-A, this arrangement causes the Computer Man to be loaded into a right-hand coordinate system as illustrated in Fig. 6-A.

The following instructions reads each slice into A(I,J) rowwise; where A(I,J) is equivalent into PLANE1(I,J).

```
      READ (4,00) NUM,MNR,MXR,MNC,MXC
  100 FORMAT(I3,2X,4I4)
      DO 2000 J=MNR,MXR
      READ(4,101)(PLANE1(I,J),I=MNC,MXC)
  101 FORMAT(5X,23I3)
 2000 CONTINUE
```

By merely reversing the order in which row entries are stored in PLANE1(I,J) the Computer Man can be assembled in a left-hand coordinate system. If the latter is chosen the instructions required to do this would be as follows:

```
      DO 2000 J=MNR,MXR
      DO 2000 I=MNC,MXC
      L=50-I
```

```
2000  READ(4,101)(PLANE1(L,J))

 101  FORMAT(5X,23I3)
```

C. Disc Description

The Ballistic Research Laboratory's Electronic Scientific Computers (BRLESC I, II) have the capability to store large quantities of data on a direct access disc. The advantage that disc storage offers is random access, in that selected portions of information can be accessed directly without reading over extraneous information which in turn results in less computer run time and improved program efficiency. Tapes are a form of sequential access and are fine for permanent storage but not for computational purposes. The user is assigned a private disc pack which in itself has a mass storage capability in excess of 800,000 words. The surfaces of the disc are divided into successive bands or tracks each of which consist of 398 words. For software purposes the disc can be divided into files for storage of separate programs or in our case, separate target descriptions, and each file can be subdivided into records. The length of each record cannot exceed 65,000 words and all records must be of equal length.

On account of the large size of the incapacitation model and the number of separate time frames required for the lethality model, more than one disc pack is required to file the descriptions.

Transferring the Computer Man description onto disc requires the definition of lethality arrays consisting of (50x35x82) entries for each of the four time frames stipulated. To work within the realm of 65,000 words of storage per disc record, the Computer Man was divided into three records with 28 slices/record. This generates 147,000 words of disc storage (50x35x84) and incorporates two additional (dummy) slices in the last record.

For incapacitation, (110x55x82) entries are required. To store a target description of this size requires in the order of a half million words. The number of slices per record must be lessened to accommodate a larger grid matrix, which in turn increases the number of records required. The choice of 14 records, allotting six slices/record, yields a total of 84 slices (82 real + 2 dummy).

To generate disc storage for the Computer Man the following specification statements must appear in the disc load routines:

```
INCAPACITATION:                          LETHALITY:

DATA DISC2/10H55D22 CMAN/                DATA IDD/10H50D21 BODY/

DATA FILE2/10HINCAP BODY/                DATA IDF/10HBODY      /

COMMON MAN2(110,55,6) NAME2(4)           COMMON MAN (50,35,28) NAME(4)
```

The following subroutine opens the disc and must appear before any other direct access disc subroutine is used.

```
CALL DISCDL (IDSC, IAVL, ICW, 'NEW DISC')
```

(1) IDSC is the I/O system unit number.

(2) IAVL is the address where the program disc label is stored.

(3) ICW is the name given to the user's disc pack.

(4) 'NEW DISC' is necessary for the initial run whenever a new disc pack is used; it must be removed from the argument list in succeeding runs.

```
INCAPACITATION:  CALL DISCDL(3,NAME2,DISC2,'NEW DISC')

LETHALITY:       CALL DISCDL (3,NAME,IDD,'NEW DISC')
```

Before reading or writing can begin the disc files must first be defined.

```
CALL DISCDF(IFIL,LOR,NT,IAV,K)
```

(1) IFIL - the name given to the file.

(2) LOF - length of each record in the file.

(3) NT - number of disc tracks required.

(4) IAV - variable associated with IFIL (dummy argument).

(5) K - flag which will permit or prevent reading or writing from unwritten records.

```
INCAPACITATION:  CALL DISCDF (FILE2,36300,1288,IDUMMY,0)

LETHALITY:       CALL DISCDF (IDF,49000,372,IDUMMY,0)
```

In order to read or write on the disc the following subroutines are used:

```
CALL DISCRD (IFIL,IF,NR,A)

CALL DISCWT (IFIL,IR,NR,A)
```

(1) IFIL - the name given to the file.

(2) IF - the record number where reading or writing is to start.

(3) NR - the number of consecutive records to transmit.

(4) A - the memory address where reading or writing begins.

INCAPACITATION: CALL DISCDF (FILE2,RECORD,1,MAN2)

CALL DISCWT (FILE2,RECORD,1,MAN2)

record varies (1 - 14)

LETHALITY: CALL DISCRD (IDF,RECORD,1,MAN)

CALL DISCWT (IDF,RECORD,1,MAN)

record varies (1 - 3)

In cases where information is transferred between two separate disc packs the system subroutine DISCSU is needed and must precede each disc read or write where a change in disc packs occurs.

CALL DISCSU (IDISC)

where IDISC is the system unit number for that particular disc.

V. COMPUTER MAN PROGRAMS - LETHALITY

A. Disc Load and Average Score Programs

1. Discussion. The Disc Load program loads the lethality description of the Computer Man onto the disc from tape and prints out the disc description of the Computer Man in its numeric representation for each doctor's evaluation given for a specified time frame.

Slice descriptions provided on cards and tape are arranged as shown in Fig. 10. With respect to time frames that require additional assessments, those assessments would follow in the order given. Fig. 11 illustrates the disc layout of the Computer Man after splicing and grafting of slices is performed. The disc load program is comprised of subroutines used to assemble, store and output the target description. A description of each subroutine used is given. The system subroutines utilized on BRLESC are not listed in this report but can be referenced in the BRLESC FORTRAN IV manual.[3] Flow charts and programs complement the documentation and are included in Appendix D.

The Average Score program averages the individual doctors' evaluations for each Computer Man cell and loads the average scores back onto the disc. Subroutines (UNLOAD and OUTPUT) listed in this section perform the essential functions of the program.

---

3. *W. Lloyd Campbell, Glenn A. Beck, BRLESC I/II FORTRAN, Technical Report No. 5, Aberdeen Proving Ground, MD 1970.* (AD #704343)

LAYOUT OF COMPUTER MAN (LETHALITY)

SLICE NO.	ANATOMICAL SECTION	DISC RECORD NO.
(1 - 28)	Legs	1
(29-33)	Legs	
(34-36)	Legs, Arms	2
(37-56)	Arms, Torso	
(57-64)	Arms, Torso	
(65-82)	Head	3
(83-84)	Dummy Slices	

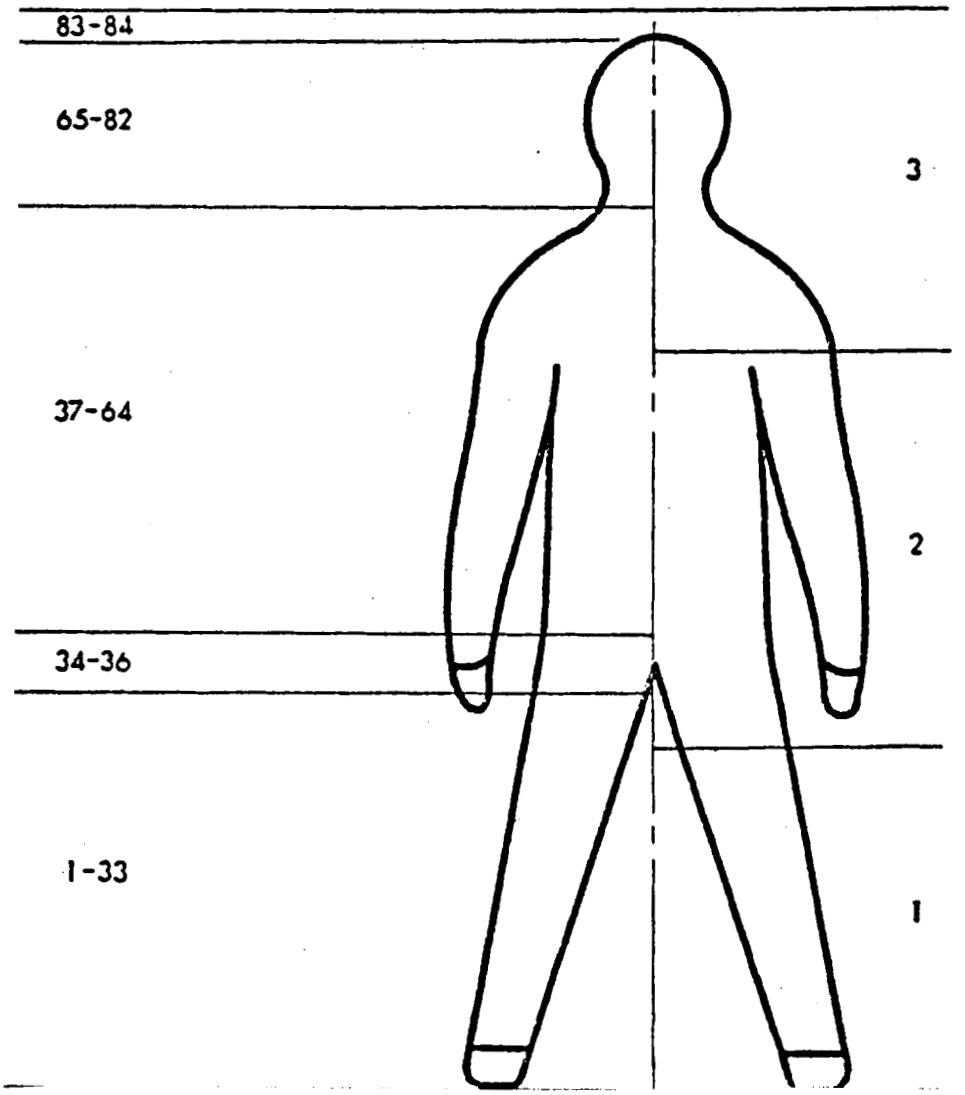


Figure 11. A Representation of the Disc Structure of the Computer Man

2. Main Routine. Definition of symbols and matrices.

M: position in subroutine (PACK,UNPACK) that is being loaded or unloaded with a score. It is synonymous to the doctors' interpretation number, M = 1,2,3

L: relative disc record number, L = 1,2,3

K: slice number within any record, K = 1-28

NDE: number of doctor evaluations provided for this time frame.

NTEST: sample selection flag. If this flag is set, the scores stored on disc in Vector (10) array are printed out for a cell with preassigned (I,J,K) coordinates.

MAN(50,35,28): target matrix in which one-third (one record of the Computer Man description for an individual assessment) is stored.

PLANE1(50,35): work array containing the slice by slice description of the Computer Man.

PLANE2(50,35): work array necessary for temporary storage of scores.

VECTOR(10): array utilized by subroutines (LOAD,UNLOAD) for storage of unpacked scores.

PROCEDURE: Initially this subroutine erases each record file before disc writing begins. The cell description of the Computer Man is transferred from tape to disc for each doctors' interpretation (M) and proceeds as follows: The head, neck and upper portion of the torso, slices (1-26), are read into active memory (PLANE1(50,35)) via (SUBROUTINE INPUT). Each integral score in each slice description is packed into the (M)th position of a word (SUBROUTINE LOAD) and this word is stored as an entry in the array MAN(50,35,28). The first 26 slices are transferred to disc and stored in the highest record on file. This action provides for two dummy slices required to complete the third section.* The initial portion of the second section, slices (27-44), is loaded onto disc in the same fashion as section one was loaded. The remaining slices, slices (50-113), are descriptions of the Computer Man's arms and legs and are dealt with in a different fashion. At this point the partial disc description of the Computer Man is read back into core, one section at a time, beginning with the highest record.

the terms section, record and cube are synonymous in useage.

The arm and leg slices which are present in each section are read in from tape, reflected (SUBROUTINE REFLECT1,REFLECT2), to generate an identical set of scores relating to the opposite arm and leg, and finally attached to the body segments that appear in the same section (SUBROUTINE GRAFT). The section currently loaded in core is written back onto disc and the section that follows is read in. This is done for the three sections that contain the man. The above procedure is repeated until all doctor evaluations have been loaded.

It is important to understand that the highest disc record (3) must be loaded first, followed by disc record (2) and disc record (1) respectively. This is necessary to ensure that the direction in which indexing is conducted in the output subroutine (Z=1,2,3,...84) is in agreement with the manner in which the Computer Man is positioned in the xyz coordinate system.

3. Subroutine LOAD. This subroutine packs all doctor's scores pertaining to a particular cell into one word and stores this word as an entry in the array MAN(I,J,K). This is done for the three sections required to describe the Computer Man.

4. Subroutine GRAFT. This subroutine is used to attach the Computer Man's arm and leg slices to their adjoining segments of the body. It can graft any slice onto another slice. If an overlap between body segments occurs on any integrated slice, the more critical score is kept.

5. Subroutine INPUT. This subroutine reads the card description of the Computer Man from tape and places it in active memory in a work array PLANE1(50,35) for manipulation and assignment onto disc. When reading is completed the current slice number (NUM) is printed out. The arguments involved in reading are defined as follows:

(1) NUM = slice or plane number.

(2) MRX = maximum number of rows for this slice.

(3) MNR = minimum number of rows for this slice.

(4) MXC = maximum number of columns for this slice.

(5) MNC = minimum number of columns for this slice.

(6) PLANE1(50,35) contains the doctors' evaluation for each cell on a given slice.

6. Subroutine REFLECT1. This subroutine reflects the left arm slices to generate right arm slices. The y axis and the line X=50 are used as reference points. It is called each time a new arm slice is read in.

7. Subroutine REFLECT2: This subroutine reflects the left leg slices to generate right leg slices. The y axis and the line X=50 are used as reference points. It, also, is called each time a new leg slice is read in.

8. Subroutine UNLOAD. This subroutine unpacks the Computer Man description and is used in conjunction with subroutine OUTPUT whenever a Computer Man printout is required. All doctors' evaluations assigned to each individual cell are stored and packed in one word on disc. The packed scores associated with each cell are expanded into an array (VECTOR) with one evaluation per word. The scores presented in this aray are averaged (rounded to the nearest integer) and this average score is stored in the tenth position. At this point, if NTEST is set, each score in VECTOR array with cell coordinates (15,15,15) are printed out in order to check individual scores and verify averaging calculation. The scores in VECTOR are repacked into one word and this entry is loaded back into MAN(50,35,28). Subsequently, the main routine will transfer both the individual and average scores back onto disc. The average scores, being representative of all doctors' assessments, are reserved for utilization in subroutine OUTPUT.

Space is provided for assignments of up to nine doctors' evaluations for each cell defined. The only programming change required is to update a data card listed in the main program, /NDE/(number of doctors' evaluations). This change will calculate the correct average score if the number of doctors' evaluations increase. The Computer Man configuration is constructed in three dimensions in this fashion for each cross section of the human anatomy defined. This subroutine is called three times since the target description is assigned to three records.

9. Subroutine OUTPUT. This subroutine outputs the disc description of the Computer Man in numeric representation on an xy plane for each slice defined. The target description is stored on disc in expanded rather than compressed format as is on tape. Thus words used to indicate air space internal to the Computer Man box are included and scored as zeros.

Also, this subroutine shifts the data such that the scores can be printed out in the range indicated in Table II.

Table II

Value Stored on Disc	Computer Man Listing
0	Blank
2	1 → minimum score
3	2
4	3
5	4
6	5
7	6
8	7
9	8
10	9
11	0 → maximum score

This can be accomplished easily if the converted scores are translated into their character representation and printed out under A2 format. Care must be utilized in calculating (K), the pointer to the numeric list where character entries are stored. If erroneous data is transferred onto disc due to missed punched cards, an exceedingly large index could result which would point to a location outside the boundary of the character array resulting in erroneous output. Therefore it is essential to check the value of K to ensure that it is within limits.

It is important to note that in this subroutine for the Computer Man to be assembled in its correct perspective in reference to the xy axis, all DO loops that control output must decrement instead of increment. This is done to compensate for the fact that the lineprinter cannot start printing at the bottom of a page and work up, only from the top and work down.

B. Change Value Program

1. Discussion. The change value program was written to modify entrees in the disc description. It should be used when the number of scored cells that require modification are relatively few. It is a convenient tool, particularly in cases where doctor interpretations are missing (principaplly with cells comprising the perimeter of the torso) and where erroneous data is detected. The maximum number of points that this program can accommodate is 10; but the program can be easily

modified to accommodate a larger number of cells. If, however, it is desired to modify entire slices it is more practical to make the necessary corrections in the card description and rerun the lethality disc load program.

2. Main Routine. Inputs required:

MAXPTS : total number of points to be modified.

DOCINT(10): doctor interpretation numbers.

IREC(10) : relative record numbers (1-3).

ISLICE(10): slice numbers (1-82); i.e., K-coordinates.

NEWVAL(10): value of new scores (1-10).

IX(10) : I-coordinate associated with each modified score.

IY(10) : J-coordinate associated with each modified score.

The input parameters are specified in data statements; as such read and format statements are not necessary.

Matrices required:

MAN(50,35,28): Computer Man section currently loaded.

PLANE2(50,35): work array associated with original slice.

PLANE(50,35) : work array that contains updated value.

A list of program instructions to be executed for each score requiring modification is given as follows:

(1) utilizing data input, read in correct disc record.

(2) store slice number, doctor interpretation number, and (I,J) coordinates for subsequent use.

(3) unpack doctors score from MAN(I,J,K).

(4) store in output buffer (PLANE(I,J)).

(5) print out slice description with original score.

(6) increment updated score.

(7) Print out slice description with updated score.

(8) Load target matrix, MAN(I,J,K) back onto disc.

When the loop, (DO 100 N=1, MAXPTS) has been completed all scores input have been modified. Program flow charts and listings are provided in Appendix D.

3. Subroutines. The subroutines used in this program (LOAD, UNLOAD, OUTPUT) are duplicates of the subroutines used in the lethality disc load program and therefore will not be discussed here.

C. Tape Translation Program

1. General. The purpose of the tape translation program is threefold. It converts the tape image of the card description of the Computer Man into a more acceptable form for disc storage. It tests first card parameters for consistency and makes corrections, when errors are detected and can be corrected. It contains provisions which allow corrections to be input by the programmer.

2. Inputs. The following correction cards apply when corrections are to be input by the programmer. If corrections are necessary, set CORRECTION = . TRUE .. A single correction card will be included for each line that requires a change. If it is necessary to change a first card parameter, then two cards are required as outlined below.

CARD 1 INDEX1, INDEX2, ITEM, JUMP, INDEX3

Format (5I5)

INDEX1 - the slice number to be changed.

INDEX2 - argument index number on the score card.

ITEM - the new value it is to assume.

JUMP - the line number relative to the previous first card, i.e., first card = 1.

INDEX3 - action flag.

INDEX3=8 - change a single entry.

INDEX3=9 - insert a new first card described by the next card.

CARD 2 ROW(I),I=1,8

Format (I3,I2,4I4,4X,2I4)

INDEX3=10 - apply no more corrections

3. Definitions of Variables and Matrices.

ROW(I),I=1,25 - this array contains slice data.

If the card read is a first card then the following is true:

ROW(1) - slice number (1-113)

ROW(2) - (-1)

ROW(3) - starting row number

ROW(4) - finishing row number

ROW(5) - starting column number

ROW(6) - finishing column number

ROW(7) - number of columns

ROW(8) - number of rows

ROW(9) - unused

.
.
.

ROW(25) - unused

If the card read is a portion of the cell description then the following is true:

ROW(1) - slice number (1-113)

ROW(2) - 1 or 2

ROW(3) - lethality score

.
.
.

ROW(25) - lethality score

NEWROW(I),I=1,8 - this array contains the first card entries input by the programmer.

KARD - the number of lines (score cards) read for a given cross section.

NUMCARDS - number of even cards for a given row description (an even card consists of a fill 23 entries).

RES - number of left over entries for a given row description (1-22).

CORRECTION - correlation flag indicator. If corrections are to be entered CORRECTION is set to TRUE.

FLAG - correction completion flag. When FLAG = TRUE all correction cards have been read.

4. Procedure. Each slice description is read in from the input tape containing the card description of the Computer Man. Tape reading is performed on a line by line basis inasmuch as there is a one to one correspondence between cards in the card file and lines in the tape file.

The status of CORRECTION is an input parameter. FLAG is initialized FALSE in the program.

The program prints the heading for the output listing in the following format.

'ROW AND COLUMN ARE A CHECK ON THE PARAMETERS THAT POSITION THE SUB-MATRIX WITHIN THE SLICE'

'THIS PROGRAM EDITS THE FIRST CARD PARAMETERS'

If changes in the slice arguments or cell description are to be made, correction cards are read in at this point.

The positioning parameters on each first card entry are tested by using the following set of algebraic equations.

TEMP = ROW(8)-ROW(4)+ROW(3)-1 (row count).

TEMP = ROW(7)-ROW(6)+ROW(5)-1 (column count).

The error (TEMP) of each row and column count is calculated and printed out in the following manner:

'SLICE' K 'ROW COUNT OFF BY' TEMP.

'SLICE' K 'COLUMN COUNT OFF BY' TEMP.

Where K is equal to the slice number specified in ROW(1) and TEMP is the amount (more or less), the calculated row and column counts differ from the values specified in ROW(7) and ROW(8). If TEMP is negative,

the finishing row or column must have the absolute value of the error added to it; if positive, the difference is subtracted from it. This feature is important inasmuch as the positioning parameters control how the Computer Man is stacked. An error indicates the possibility that a row or column would be shaved off the cross section when it is read into A(I,J). The finishing row and column parameters are corrected, if in error. The number of even cards (NUMCARDS) are calculated for a given row in the slice description along with the number of left over entries (RES). NUMCARDS and RES are constant for all rows defined on a given slice. At this point a check is made to see if any default conditions exist. Default conditions are defined as follows:

(1) ROW(2) $\neq$ (-1).

(2) The calculated number of score positions does not agree with the specified number given in ROW(7).

A default condition will print out an error message in the following format and terminate execution of the program.

ITEM: xxxERRORxxxCARD CHECK ___

COMPUTED NUMBER OF ENTRIES ___

INPUT NUMBER OF ENTRIES ___ ON SLICE ___

COMPUTER NUMBER OF CARDS ___ WITH ___ LEFTOVER

If a default condition does not exist the following message is printed indicating that the slice arguments are acceptable.

'NUMBER OF EVEN CARDS PER ROW (NUMCARDS) WITH RES LEFT OVER'

The program proceeds to read in the succeeding scores associated with the first card parameters in question. If a correction is indicated for a score positioned on the line currently being read, it is inserted at this point.

Each score is converted from its character representation to its integer representation via (DECODE) and incremented. This shifts the range of scores to (2-11). Also, it is necessary at this point to remove each 'B', otherwise when read into BRLESC memory a -23 would appear in the word position where each 'B' occurred causing difficulties in loading the cell description onto disc. Therefore 'B' is converted to 0. As each line is completed the converted version is written onto the output tape.

The output tape should be tabbed fully to check for discrepancies. If errors are detected, a set of correction cards should be made and the program should be run with CORRECTION equal .TRUE..

D. Computer Man Delineation Program

1. General. There are many instances in personnel vulnerability studies where our interest is focused around vital organs or regions of the body apart from the body as a whole. An example is the design of helmets where improved protective characteristics could ultimately reduce casualties resulting from head injuries. In applying the Computer Man modeling technique, we would only be concerned with the rays which penetrate the head slices of the Computer Man. This program was written for this purpose. It delineates any desired region of any given dimensions inside the body box and loads only the delineated region onto a new disc file. A modification of the ray tracing program (RAYMAN) will assess the disc file containing the modified Computer Man description. Any number of sections can be used to approximate the desired region with two restrictions: (1) the coordinates chosen for each section must define the vertices of a rectangular box, and (2) in generating each section:

$I2 \geq I1$

$J2 \geq J1$

$K2 \geq K1$

2. Inputs.

CARD 1 NUM, ISECT, ISL

NUM - the number of rectangular blocks required to construct the delineated region.

ISECT - the number of disc records which comprise the Computer Man.

ISL - the number of slices per record.

Format (3I3)

CARD 2 I1, I2, J1, J2, K1, K2

. .
. .
. .

CARD (NUM+1)

I1 - minimum x coordinate

I2 - maximum x coordinate

J1 - minimum y coordinate

J2 - maximum y coordinate

K1 - minimum z coordinate

K2 - maximum z coordinate

Format (6I3)

Note: for NUM > 1 the cards must be stacked in ascending order with respect to z coordinates.

3. Arrays and Variables.

KMAX - the last delineated slice in a section.

KMIN - the first delineated slice in a section.

NSLICE(N),N=1,14 - each location in this array contains the number of delineated slices present in each section.

KNUM(N),N=1,14 - each location in this array contains the address (slice number) where delineation starts relative to the beginning of each section. It is synonymous in meaning to KMIN.

KMR1(N),N=1,84 - this array is used to designate which slices of the Computer Man are delineated.

KMR2(N),N=1,84 - this array is used to designate where section boundaries occur.

MAN1(I,J,K) - Computer Man array.

4. Main Program. Initially the arrays KNUM, NSLICE, KMR1, KMR2 and the disc file reserved for the modified Computer Man description are erased. The program reads the input parameters defined on Card 1 which defines the upper limit (NUM) for indexing delineated regions. The program reads in the card containing the coordinates for the first delineated section. The contents of KNUM, NSLICE, KMR1, KMR2 are calculated (Subroutine CALSCE) and printed out. The cells which fall within the region defined by I1, I2, J1, J2 inclusive are loaded into MAN1(I,J,K) for each slice that is delineated within the section currently in core. The maximum and minimum values of K (KMAX, KMIN) which define the slice boundaries for each section are retrieved from KNUM and NSLICE. All DO loops will be executed once if the upper and lower limits are equal. The record currently loaded in core is transferred to its proper location in the new disc file.

The procedure is repeated until the number of input cards read containing section boundary coordinates is equal to the number of defined sections (NUM). A slice by slice description of the delineated version of the Computer Man model is printed out.

5. Subroutine CALSE. This subroutine sets slice markers in KMR1 and section boundary markers in KMR2. The number of delineated slices (KCOUNT) which are present in each section within the region defined by K2-K1 are counted and the relative address in each section (KN) where the first delineated slice occurs is stored.

6. Subroutine INSERT. This subroutine inserts the values (KCOUNT and KN) calculated in subroutine CALSCE in the arrays KNUM and NSLICE. KSECT points to the position in KNUM and NSLICE where KCOUNT and KN are stored. KCOUNT and KN are initialized before tracking begins in the next section. This subroutine is called each time a section crossing occurs provided the sections fall within the delineated region.

E. Instructions for Using Lethality Programs

1. Generate a card file as described in IV-A.

2-A. Load the card file onto tape.

2-B. Tab the tape version of the card description and correct errors in the card file.

2-C. Reload card file onto tape.

3-A. Using the tape version of the card description as input and a second tape as output, run the tape translation program with the correction status flag set.FALSE..

3-B. Tab out tape fully. If errors are detected, rerun program with correction cards inserted and the correction status flag set .TRUE..

4. With the tape containing the translated version of the card description as input, run the disc load program. Examine individual doctor's evaluations.

5. If errors exist, run the change value program to correct errors in evaluations.

6. Run average score program.

7. If an examination of body parts is required, run Computer Man delineation program.

VI. COMPUTER MAN PROGRAMS - INCAPACITATION

The programs which comprise the incapacitation section involve translating the incapacitation model into a form analogous to the lethality model and redefining the disc file structure to reduce disc access time.

The incapacitation section consists of the following three programs, Model Repositioning, Foot Slice Addition, and Mosaic Printout.* The flow charts and listings associated with these programs are listed in Appendix D.

A. Model Repositioning

1. Discussion. Unlike the lethality model which was generated in-house, the incapacitation model was generated earlier and was based upon the casualty criteria described by Allen and Sperrazza.[4] The source of the model was a tape consisting of codes which identified anatomical tissues. These codes were organized on tape in accordance with the tape description generated for lethality as outlined in Section IV. The original program which assembled and loaded the incapacitation man onto disc positioned the Computer Man standing in a left-hand coordinate system. This program repositions the Computer Man to match that of the lethality model (right-hand coordinate system) and in doing so enable the model to interface with the shotline trace program (RAYMAN) developed by BRL.[5] In addition the file and record lengths were redefined to reduce the I/O required by RAYMAN.

2. Definition of Matrices.

WORK(55,110) - work arrays associated with original model
WORK1(55,110) - required for manipulation and storage of codes.

MAN1(55,110) - target matrix associated with original model.

WORK3(110,55) - work array associated with reconstructed model.

The "Model Repositioning" and "Foot Slice Addition" programs are special purpose programs; therefore, if future incapacitation models were to be constructed using the method defined for lethality in Section IV, they would not be required.

4. *F. Allen and J. Sperrazza, "New Casualty Criteria for Wounding by Fragments, Ballistic Research Laboratories Report No. 996, 1956. (CONF)* (AD #137681)

5. *W.B. Beverly, "RAYMAN: A FORTRAN Computer Code for Tracking Rays Through a Detailed Human Phantom," USA ARRADCOM Ballistic Research Laboratory Report No. ARBRL 2030, Nov 1977.* (AD #A051057)

MAN(55,110,6) _ target matrices associated with reconstructed
MAN2(110,55,6) model.

3. Procedure. The original cell description is read in from disc and stored in MAN1(55,110). Each slice description is read individually because each slice was assigned separate records in the original model. A new work array WORK1(55,110) is loaded with the cell descriptions contained in MAN1; in loading this array indexing is conducted in the J dimension in reverse order. Next, WORK2(55,110) is loaded from WORK1(55,110) with indexing conducted in the I dimension in reverse order. Reversing the loading order of tissue codes in both the I and J dimensions, in effect, faces the Computer Man in the opposite direction.

The second step required is to invert the Computer Man. This is done by reversing the stacking order. Concurrently, the slice descriptions are regrouped and buffered in the array MAN(55,110,6) where 55x110x6 defines the storage capacity required for one record in the translated version. The following chart relates the disc file structure of the reconstructed and original models.

Table 3

DISC FILE CONVERSION CHART

	RECORDS					
FILE 1	73-76	... 19-24	13-18	7-12	1-6	
FILE 2	1	10	11	12	13	14

Note: As illustrated the new model is composed of 14 sections (records) as opposed to 76; each record consisting of six slices.

The third step required in the transformation is to rotate the Computer Man 90 degrees inside the first quadrant. This is accomplished by interchanging the (I,J) coordinates of each entry in the array MAN1(J,I,K), whereby the contents of each record assumes the form designated by MAN(I,J,K).

Each record in the new file is constructed in this fashion. However, since 82 slices represent the Computer Man, 84 defines the file, it is necessary to shift the Computer Man six slices in the -z direction after the last record is loaded. This move will incorporate two dummy slices

above the head and stand the man flush on the xy plane. Provisions are included in defining the model which will allow for six additional foot slices to be added. Also, the program prints out each slice description of the reconstructed model by calling subroutine MOSAIC. Fig. 12 illustrates the disc structure of the incapacitation model after transformation is completed.

B. Foot Slice Addition

1. Description. The original incapacitation model derived in VI-A. was constructed without feet due to the fact that cross sections (108-113) were not derived from cuts taken parallel to the xy plane. However, a soldier's mobility is of vital concern to infantry vulnerability analyses, thus it was deemed essential to consider the effect of wounds received in this region in our study.

The Foot Slice Addition program was written for this purpose. It attaches the foot slices provided by Eycleshymer and Schoemker onto the Computer Man and in effect stands him on his toes.

The program is set up such that it could be easily modified to add or delete tissue codes in any cross section.

2. Main Routine. The following instructions read in the foot slices:

```
DO 2000 SLICE = 1,6

READ (5,100) CARDNUM,ARG1,ARG2,ARG3,ARG4,ARG5,ARG6

100 FORMAT (I3,1X,6I4)

READ (5,101) ARG7,ARG8,ARG9, (ARRAY(N),N = 1,23)

101 FORMAT (I2,1X,I2,1X,I2,23I3)
```

where:

ARG1 = starting position of submatrix on y axis

ARG2 = end position of submatrix on y axis

ARG3 = starting position of submatrix on x axis

ARG4 = end position of submatrix on x axis

ARG5 = number of columns for submatrix

ARG6 = number of rows for submatrix

LAYOUT OF COMPUTER MAN (INCAPACITATION)

SLICE NO.	ANATOMICAL SECTION	DISC RECORD NO.
(1 - 6)	FEET	1
(7 - 33)	LEGS	2,- 6
(34-39)	ARMS, LEGS	6,- 7
(40-59)	ARMS, TORSO	7, - 10
(60-64)	TORSO	10,- 11
(65-82)	HEAD	11,- 14
(83-84)	DUMMY SLICES	14

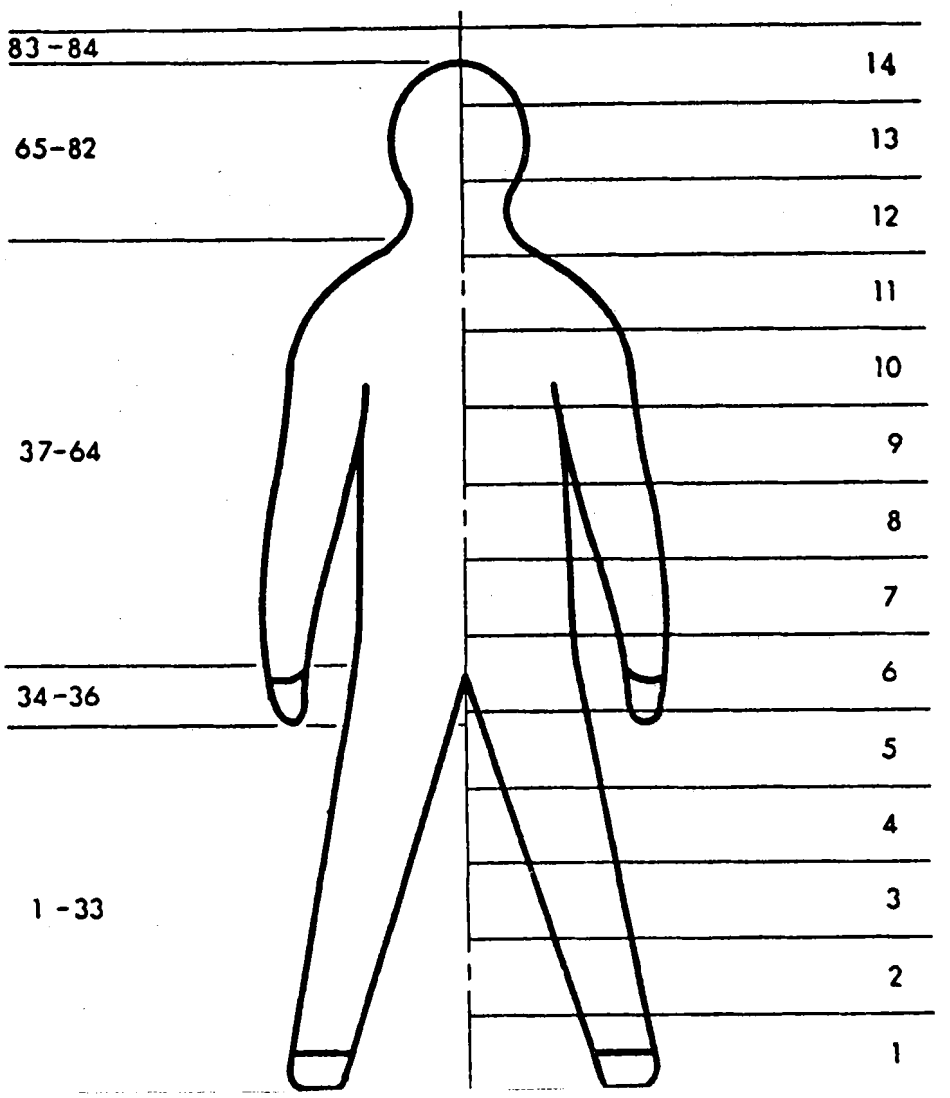


Figure 12. A Representation of the Disc Structure of the Computer Man

ARG7,ARG8,ARG9 pertain to each individual row of tissue codes.

ARG7 = y coordinate of row

ARG8 = starting x coordinate of tissue codes

ARG9 = x coordinate where tissue codes end

CARDNUM = slice number 1-113

ARRAY(23) = tissue code input list

WORK1(55,110) = temporary storage of slice description

MAN3(110,55,6) = target matrix

The tissue codes are read into ARRAY(24) rowwise in accordance with format 101.

The following steps construct and position the submatrix.

1. The entries in ARRAY are shifted upwards such that ARRAY(1) contains the first valid (non-zero) entry.

2. The tissue codes are transferred to and positioned in WORK1(55,110). The arguments ARG7, ARG8, ARG9 define the location and bounds of the submatrix.

3. The slice description is reflected about the midpoint of the x axis, line X=55, to generate tissue codes pertaining to the opposite foot.

4. The submatrix WORK1 is copied onto the main target matrix.

5. The steps outlined in the Model Repositioning Program (Section VI-A) are duplicated.

6. The foot slices are loaded onto the disc version of the reconstructed model.

7. Subroutine MOSAIC is called to print out each slice.

8. The incapacitation reconstruction program is rerun to verify that both programs interface properly.

C. Subroutine MOSAIC (KSI,I)

1. Discussion. The function of subroutine MOSAIC is to print out each cross section of the incapacitation model in its mosaic characterization. A list of the symbols which appear in the MOSAIC printout folfowed by a list of tissue codes identified by each symbol is included in Appendix A for reference.

2. Definition of Variables and Matrices.

KSI(110,55) - this array contains the tissue codes that make up the cell description of the cross section currently processed.

GRID(110,55) - this is the output array for the incapacitation model.

CODE(200) - this array contains the list of tissue codes defined to describe the Computer Man. The codes range in value (2-200).

SYMBOL(200) - the first 187 elements contain the Holerith representation of the characters used to identify tissue codes.

ORD(112) - this array is filled with 112 periods which serve as the upper and lower borders for each cross section.

N - the number of valid tissue codes defined for incapacitation (181).

I - the cross-section number (1-84).

3. Procedure. The subroutine indexes through KSI(110,55) checking for a match in tissue codes between each element in KSI(110,55) and CODE (200). Where a match is encountered, the symbol chosen to identify the tissue code is taken from SYMBOL(200) and place in GRID(110,55) in the position that corresponds to the index (I,J) in the array KSI(110,55). This is done for each element in KSI(110,55). A blank character is inserted where non-tissue codes are encountered. After each row in the output matrix GRID(110,55) has been filled, the entire slice is printed out under A2 format. Each slice description is positioned along an xy axis enclosed in a border with the cross-section number printed below the x axis. In order to assist in marking off cell distances periods are inserted at equal intervals along both axes where tissue codes do not exist.

It is important to note, as was the case with lethality, that in printing out each slice row indexing must be conducted in reverse order (i.e., ROW(55), (54), (53),...(1)) so that the Computer Man can be printed out in its correct perspective (i.e., facing parallel to and in the direction of the +y axis).

APPENDIX A

DESCRIPTION OF ANATOMICAL TISSUES WHICH DEFINE THE COMPUTER MAN

TISSUE CODE	ID SYMBOL	CROSS SECTION	ANATOMICAL STRUCTURE
2	Q	1-113	SUBCUTANEOUS
3	W	1-6	SCALP
4	E	1-12	SKULL
5	R	1-12	SKULL
6	T	1-12	BRAIN (EXCEPT FRONTAL LOBE)
23	Y	1-8	BRAIN (FRONTAL LOBE)
7	U	7-16	FACE (SOFT TISSUE)
8	I	7-16	FACE (CANC BONE)
9	Ø	7-16	FACE (CORT BONE)
10	P	7-9	EYE (EXCEPT EYEBALL AND OPTIC NERVE)
24	A	7-9	EYEBALL AND OPTIC NERVE
11	S	7-18	MUSCLE
12	D	18-20	NERVE C-5,6,7
13	F	11-19	PHARYNX
14	G	17-19	LARYNX
179	H	17	HYOID BONE
15	J	13-20	VERTEBRA (SPINAL AND TRANSVERSE PROCESSES)
16	K	13-20	VERTEBRA (SPINAL AND TRANSVERSE PROCESSES)
17	L	12-21	VERTEBRA (ARCH AND BODY)
18	Z	12-21	VERTEBRA (ARCH AND BODY)
173	X	12-21	BONE WITHIN 1 CM OF SPINAL COLUMN
174	C	12-21	BONE WITHIN 1 CM OF SPINAL COLUMN
19	V	13-21	SPINAL CORD
20	B	1-10	VASCULAR
20	B	1-8	SUPERIOR SAGGITAL SINUS
20	B	3	SUPR CEREBRAL SINUS
20	B	2-3	MIDDLE CEREBRAL SINUS
20	B	2-6,9	MIDDLE MENINGEAL SINUS
20	B	9-11	TRANSVERSE MENINGEAL SINUS
20	B	9-10	SINUS PETROSUS INFERIOR
21	N	9-18	VASCULAR
21	N	9-17	INTERIOR CAROTID
21	N	11-17	EXTERIOR CAROTID
21	N	18	COMMON CAROTID
22	M	8-18	VASCULAR
22	M	8-11	SUPERFICIAL TEMPORAL
22	M	11,13-15	OCCIPITAL
22	M	11,12	ANGULAR
22	M	12,13	POST AURIC
22	M	13-17	ANT FACIAL
22	M	14-17	EXT MAXILLARY
22	M	14-18	GENU PROFUNDAS
22	M	16,17	SUBMENTAL
22	M	18	ANT JUGULAR
22	M	18	COMM FACIAL
22	M	18	V.COMMUNICANS
22	M	17	POST. FACIAL
25	O	11-18	VASCULAR
25	O	11-18	INT JUGULAR
25	O	11,13-18	EXT JUGULAR
55	I	22-27	VASCULAR

TISSUE CODE	ID SYMBOL	CROSS SECTION	ANATOMICAL STRUCTURE
25	0	12-18	VERTEBRAL
25	0	17-18	SUPER THYROID
26	2	19-27	MUSCLE
27	3	19-27	PHRENIC NERVE
28	4	21-23	NERVE
28	4	21-23	BRACHIAL PLEXUS
28	4	21-23	ULNAR
28	4	21-23	RADIAL
28	4	21-23	MEDIAN
29	5	21-31	VERTEBRA (TRANSVERSE AND SPINAL PROCESS)
30	6	21-31	VERTEBRA (TRANSVERSE AND SPINAL PROCESS)
31	7	22-31	VERTEBRA (ARCH AND BODY)
32	8	22-31	VERTEBRA (ARCH AND BODY)
175	9	22-31	BONE WITHIN 1 CM OF SPINAL COLUMN
176	Q	22-31	BONE WITHIN 1 CM OF SPINAL COLUMN
33	W	22-31	SPINAL CORD
34	E	19-21	THYROID
35	R	20-24	TRACHEA
36	T	20-28	ESOPHAGUS
37	Y	20-21	CLAVICLE
38	U	20-21	CLAVICLE
39	I	20-25	SCAPULA
40	Ø	20-25	SCAPULA
41	P	21	SHOULDER JOINT (HEAD OF HUMERUS AND ADJ SCAPULA)
42	A	21	SHOULDER JOINT (HEAD OF HUMERUS AND ADJ SCAPULA)
177	S	21	ARTICULAR SURFACE OF SHOULDER
178	D	21	ARTICULAR SURFACE OF SHOULDER
43	F	22-23	HUMERUS (PROX SHAFT)
44	G	22-23	HUMERUS (PROX SHAFT)
45	H	21-33	RIB
46	J	21-33	RIB
47	K	21-28	LUNG
48	L	27-31	DIAPHRAGM
49	Z	25-27	HEART (EPICARDIUM AND MYOCARDIUM)
50	X	25-27	HEART (CHAMBERS)
51	C	19-21	VASCULAR
51	C	19-21	INT JUGULAR
51	C	19-21	EXT JUGULAR
51	C	20	INF THYROID
51	C	19-20	VERTEBRAL
52	V	19-27	VASCULAR
52	V	24-27	PULMONARY
52	V	25-27	CORONARY
52	V	24-27	AZYGOS
52	V	24-27	VENA CAVA
52	V	21-22	SUB CLAVIAN
52	V	19-21	COMMON CAROTID
53	B	23-27	STERNUM
54	N	22-27	VASCULAR
54	N	22-23	AXILLARY
54	N	22-27	INT MAMMARY
73	M	30	ARCHOFORMIS

TISSUE CODE	ID SYMBOL	CROSS SECTION	ANATOMICAL STRUCTURE
55	0	23-27	AORTA
55	0	22	INNOMINATE
56	1	28-32	STOMACH
57	2	28-42	INTESTINE
58	3	28-34	LIVER
59	4	32-33	GALL BLADDER
60	5	28-29	SPLEEN
61	6	29-32	PANCREAS
62	7	29-30	ADRENALS
63	8	29-33	KIDNEY
64	9	31-40	URETER
65	Q	32-37	VERTEBRA (SPINAL AND TRANSVERSE PROCESS)
66	W	32-37	VERTEBRA (SPINAL AND TRANSVERSE PROCESS)
67	E	32-37	VERTEBRA (ARCH AND BODY)
68	R	32-37	VERTEBRA (ARCH AND BODY)
75	T	32-35	VERTEBRA (WITHIN 1 CM OF SPINAL CHORD)
69	Y	32-37	VERTEBRAL CANAL
70	U	28-35	MUSCLE
71	I	28-35	VASCULAR (AORTA)
72	Ø	28-35	VASCULAR
72	Ø	28-30	HEPATIC
72	Ø	29-32	RENAL
72	Ø	31	GASTRODUODENAL
72	Ø	30-33	PORTAL
72	Ø	31	CYSTIC
72	Ø	31-33	LEFT COLIC
72	Ø	31-32	RIGHT GASTRIC
72	Ø	31	MIDDLE COLIC
72	Ø	31	PYLORIC COLIC
72	Ø	32-35	SPERMATIC
72	Ø	33-35	JEJENAL
72	Ø	34-35	RIGHT COLIC
72	Ø	34-35	INTESTINAL
72	Ø	35	ILEAC
72	Ø	35	ILEO COLIC
72	Ø	35	SUPERIOR HEMM
72	Ø	35	SIGMOIDAL
72	Ø	28	INT MAMMARY
72	Ø	32	PANCREATODUODENAL
73	P	28-35	VASCULAR
73	P	28-30	INF PHRENIC
73	P	28-30	INTERCOSTALS
73	P	28-30	HEMIAZYGOS
73	P	28	ESOPHAGEAL
73	P	28	SUP SUPRARENAL
73	P	28	GASTRIC BREVIS
73	P	28	PERICARDIA PHRENIC
73	P	29	INF SUPRARENAL
73	P	29-30	CORONARY VENTRICULI
73	P	29	MUSCULOPHRENIC
73	P	30-31	SUPRARENAL
73	P	30	INTERLOBULARIS
96	A	39-42	OBTURATOR

TISSUE CODE	ID SYMBOL	CROSS SECTION	ANATOMICAL STRUCTURE
73	P	31	LUMBALIS
73	P	34-35	LUMB ASC
73	P	35	ILEOLUMBALIS
74	S	28-35	VASCULAR
74	S	28-35	VENA CAVA
74	S	29-30	LEFT GASTRIC
74	S	29-31	SPLENIC
74	S	32-34	SUPERIOR MESENTERIC
74	S	32-35	INF MESENTIVIC
74	S	28-29	AZYGOS
76	D	34-44	NERVE-3,4 LUMBAR" LUMBROSACRAL 1,2,3 SACRAL., FEMORAL, OBTURATOR
77	F	40-44	NERVE SCIATIC
78	G	37-40	VERTEBRA (SACRUM)
79	H	37-40	VERTEBRA (SACRUM)
180	J	38-40	BONE (1 CM OR LESS FROM CAUDA EQUINA)
80	K	41	VERTEBRA (COCCYX)
81	L	41	VERTEBRA (COCCYX)
82	Z	35-39	VERTEBRA (WING OF ILIUM)
83	X	35-39	VERTEBRA (WING OF ILIUM)
84	C	41-43	PUBIS
85	V	41-43	PUBIS
86	B	42-43	ISCHIUM
87	N	42-43	ISCHIUM
88	M	40-41	HIP JOINT
89	O	40-41	HIP JOINT
99	1	40-41	ARTICULAR SURFACE OF HIP JOINT
100	2	40-41	ARTICULAR SURFACE OF HIP JOINT
90	3	42-44	FEMUR
91	4	42-44	FEMUR
92	5	36-44	MUSCLE
93	6	41-42	URINARY BLADDER AND URETHRA
94	7	42-44	EXT GENITALIA
95	8	36-44	VASCULAR
95	8	36	VENA CAVA
95	8	36-37	COMM ILIAC
95	8	37-40	EXT ILIAC
95	8	37-39	HYPOGASTRIC
95	8	41-44	FEMORAL
95	8	41-44	PROFUNDA FEMORIS
96	9	36-43	VASCULAR
96	9	36-38	ILEOCOLIC
96	9	36	A ILEAC
96	9	36-38	RT COLIC
96	9	36-38	SPERMATIC
96	9	36	INTESTINAL
96	9	36,37	JEJENAL
96	9	36-38	SIGMOIDAL
96	9	36-39	SUPRAHEMMORCIDAL
96	9	39	SUPRA GLUTEAL
96	9	40-43	INF GLUTEAL
119	Q	60-66	VASCULAR

TISSUE CODE	ID SYMBOL	CROSS SECTION	ANATOMICAL STRUCTURE
96	9	40-42	INT PUDENAL
96	9	40	INF VESICLE
96	9	36-38	MID SACRAL
97	W	36-44	VASCULAR
97	W	41-44	SAPHRENA MAGNA
97	W	36-40	INF EPIGASTRIC
97	W	36-38	ILEOLUMBALIS
97	W	37-39	CIRCUMFLEX ILEUM PROF
97	W	38-41	LAT SACRAL
97	W	41-44	PAMPINI FORM PLEXUS
97	W	42	DORSAL PENIS
97	W	42-44	LAT CIRC FEMORIS
97	W	42-44	PRIM PERFORANS
97	W	43	PROF PENIS
97	W	43	PUDENAL PLEXUS
98	E		
101	R	50-59	MUSCLE
102	Y	50-59	NERVE - MUSCULOCUTANEOUS ULNAR, MEDIAN, RADIAL
103	U	50-57	HUMERUS
104	I	50-57	HUMERUS
105	Ø	58-59	ELBOW JOINT
106	P	58-59	ELBOW JOINT
110	A	58-59	ARTICULAR SURFACE OF ELBOW JOINT
111	S	58-59	ARTICULAR SURFACE OF ELBOW JOINT
107	D	50-59	VASCULAR
107	D	50-59	BRACHIAL
107	D	50-51	PROFUNDA BRACHII
108	F	50-59	VASCULAR
108	F	50-59	CEPHALIC
108	F	50-59	BASILIC
109	G	52-59	VASCULAR
109	G	52-54	SUP ULNAR COLLATERAL
109	G	52-53	MIDDLE COLLATERAL
109	G	52-55	RADIAL COLLATERAL
109	G	58-59	ULNAR RECURRENT
109	G	58-59	RADIAL RECURRENT
109	G	59	INF. ULNAR COLLATERAL
112	H		
113	J	60-68	MUSCLE
114	K	60-68	NERVE - ULNAR, RADIAL, MEDIAN
115	L	60-68	RADIUS AND ULNA
116	Z	60-68	RADIUS AND ULNA
117	X	60-68	VASCULAR
117	X	60-68	RADIAL
117	X	60-68	ULNA
118	C	60-70	VASCULAR
118	C	60-70	BASILIC
118	C	60-67	CEPHALIC

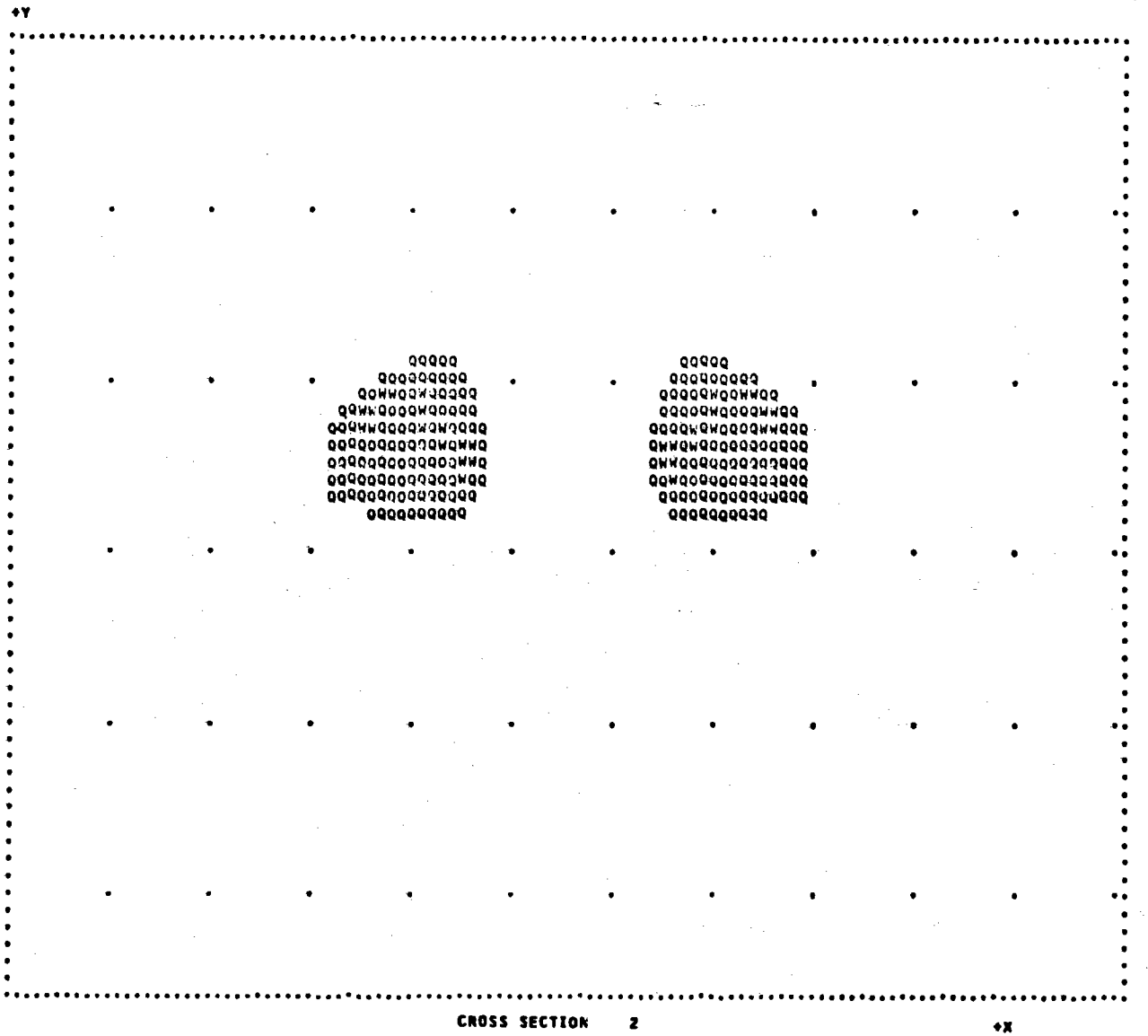
TISSUE CODE	ID SYMBOL	CROSS SECTION	ANATOMICAL STRUCTURE
119	V	60	RADIAL RECURRENT
119	V	60	ULNAR RECURRENT
119	V	61	COMMON INTEROSSEUS
119	V	62-66	VOLAR INTEROSSEUS
119	V	62-66	DORSAL INTEROSSEUS
120	B		
121	N		
122	M		
123	O	69-71	MUSCLE
124	1	69-71	NERVE - ULNAR, MEDIAN
125	2	69	RADIUS AND ULNA
126	3	69	RADIUS AND ULNA
127	4	70-71	CORPAL BONES
128	5	70-71	CORPAL BONES
129	6	69-71	VASCULAR (ULNAR AND RADIAL)
130	7		
131	8		
132	9	72-75	SOFT TISSUE
133	Q	72-75	BONE
134	W	72-75	BONE
135	E		
136	R	76-92	MUSCLE
137	T	76-86	NERVE - SCIATIC
138	Y	87-92	NERVE - TIBIAL, PERONEAL
139	U	76-87	FEMUR
140	I	76-87	FEMUR
141	Ø	88-92	KNEE
142	P	88-92	KNEE
146	A	89-91	ARTICULATING SURFACE OF KNEE
143	S	76-92	VASCULAR
143	S	76-85	TEMPORAL
143	S	86-92	POPLITEAL
143	S	76-83	PROFUNDA FEMORAL
144	D	76-92	VASCULAR
144	D	76-92	SAPHENA MAGNA
144	D	86-92	SAPHENA PARVA
145	F	76-92	VASCULAR
145	F	76-92	PRIMARY PERFORATING
145	F	76-92	SECONDARY PERFORATING
145	F	76-92	TERTIARY PERFORATING
145	F	76-92	GENU SUPREMA
145	F	76-92	GENU INF. MED.
147	G		
149	H	93-105	MUSCLE
150	J	93-97	NERVE - TIBEAL, PERONEAL
151	K	93-105	TIBIA
152	L	93-105	TIBIA
153	Z	93-105	FIBULA
154	X	93-105	FIBULA

TISSUE CODE	ID SYMBOL	CROSS SECTION	ANATOMICAL STRUCTURE
155	C	93-105	VASCULAR
155	C	93	POPLITEAL
155	C	94-105	POST TIBIAL
156	V	94-105	VASCULAR
156	V	94-104	PERONEAL
156	V	94-105	ANT. TIBIAL
157	B	93-105	VASCULAR (SAPHENA MAGNA
158	N	106-109	VASCULAR
159	M		
160	O	106-109	MUSCLE
161	1	106-107	TIBIA
162	2	106-107	TIBIA
163	3	106	FIBULA
164	4	106	FIBULA
165	5	107-109	TARSAL BONES
166	6	107-109	TARSAL BONES
167	7	106-109	VASCULAR
167	7	106-107	ANT. TIBIAL
167	7	106-107	POST. TIBIAL
167	7	108-109	DORSALIS PEDIS
168	8	106-109	VASCULAR
169	9		
170	Q	110-113	SOFT TISSUE
171	W	110-113	BONE
172	3	110-113	BONE

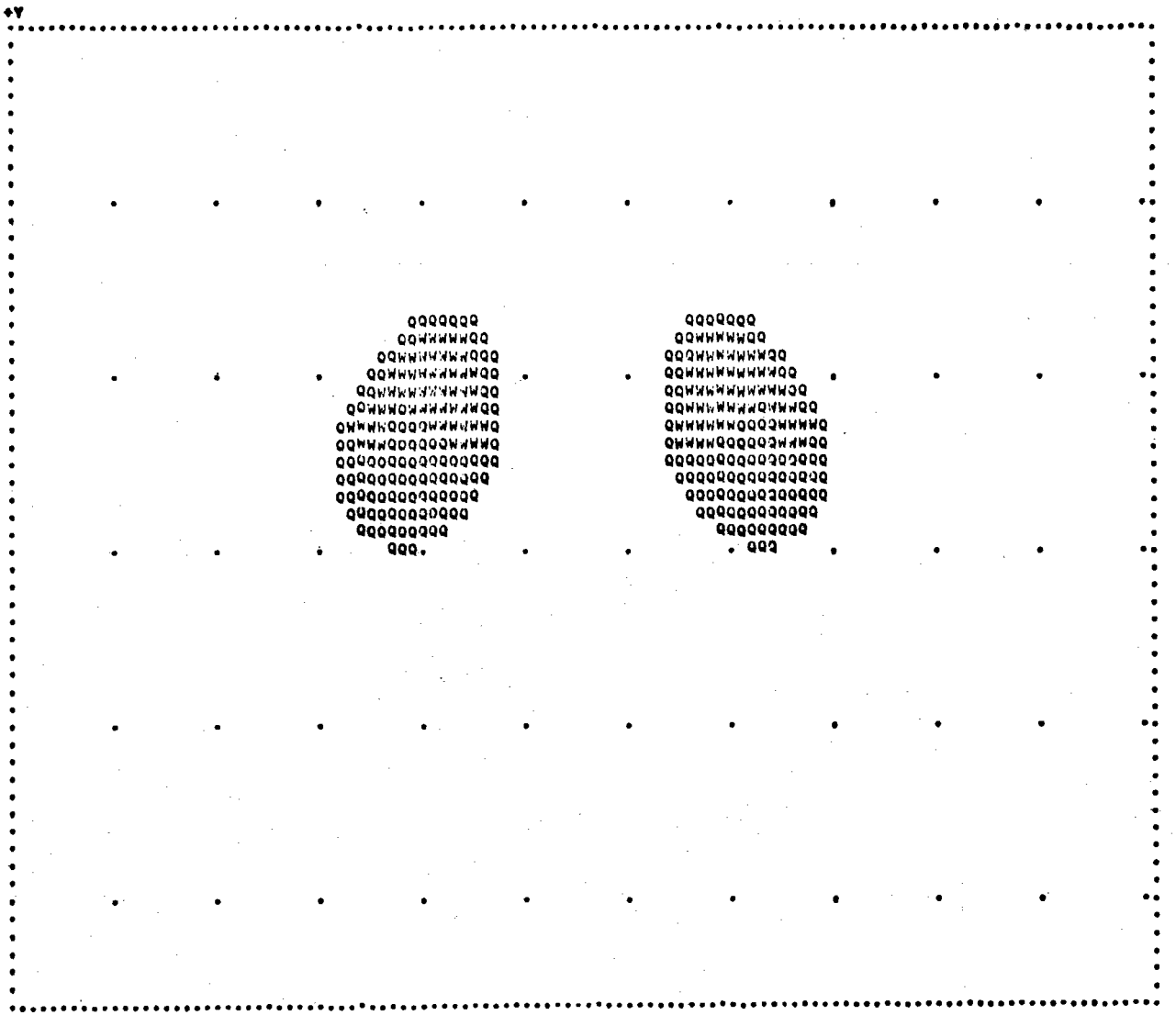
APPENDIX B

A SLICE BY SLICE COMPUTER DESCRIPTION OF THE INCAPACITATION MODEL

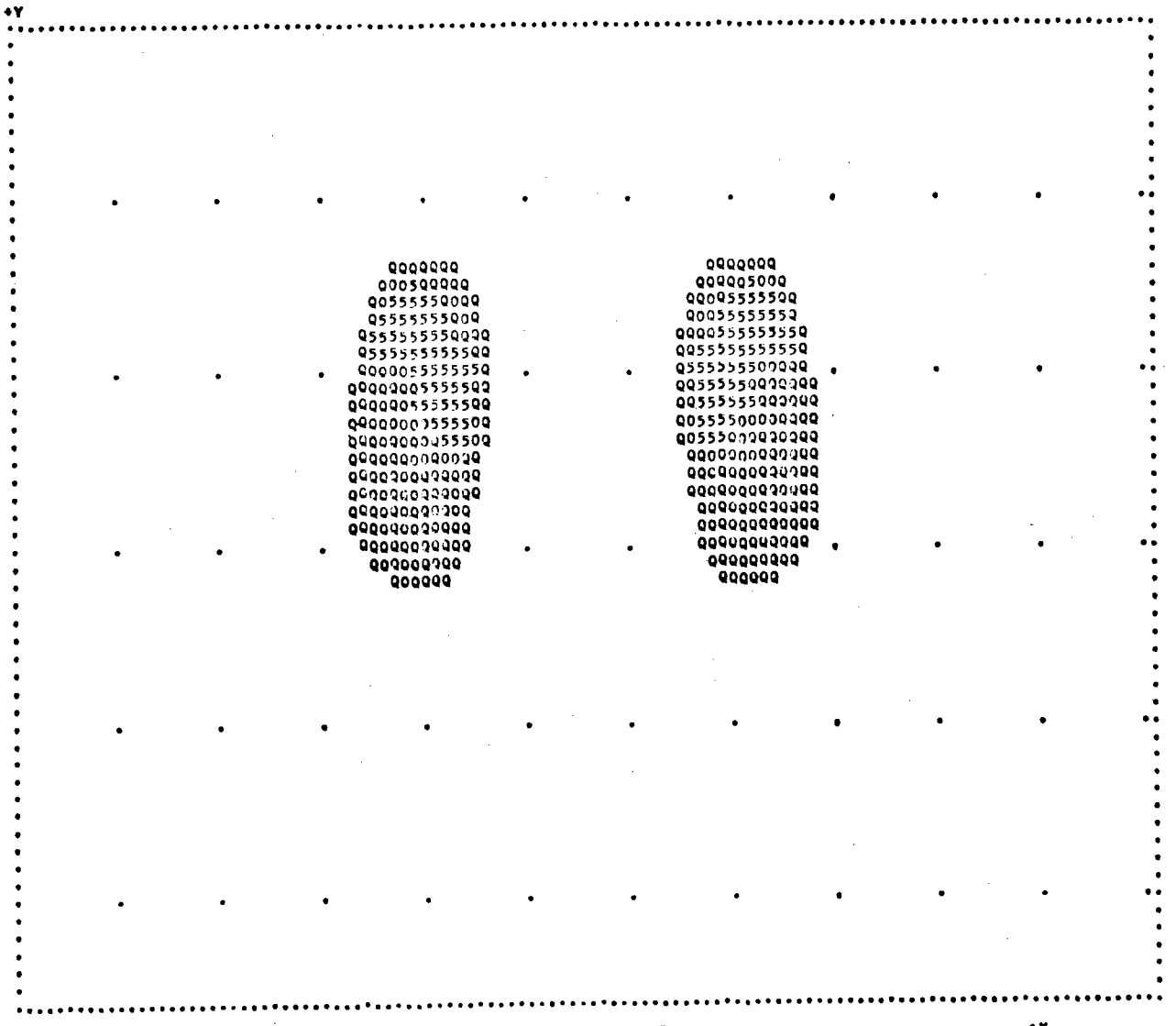
+Y
CROSS SECTION 1
+X

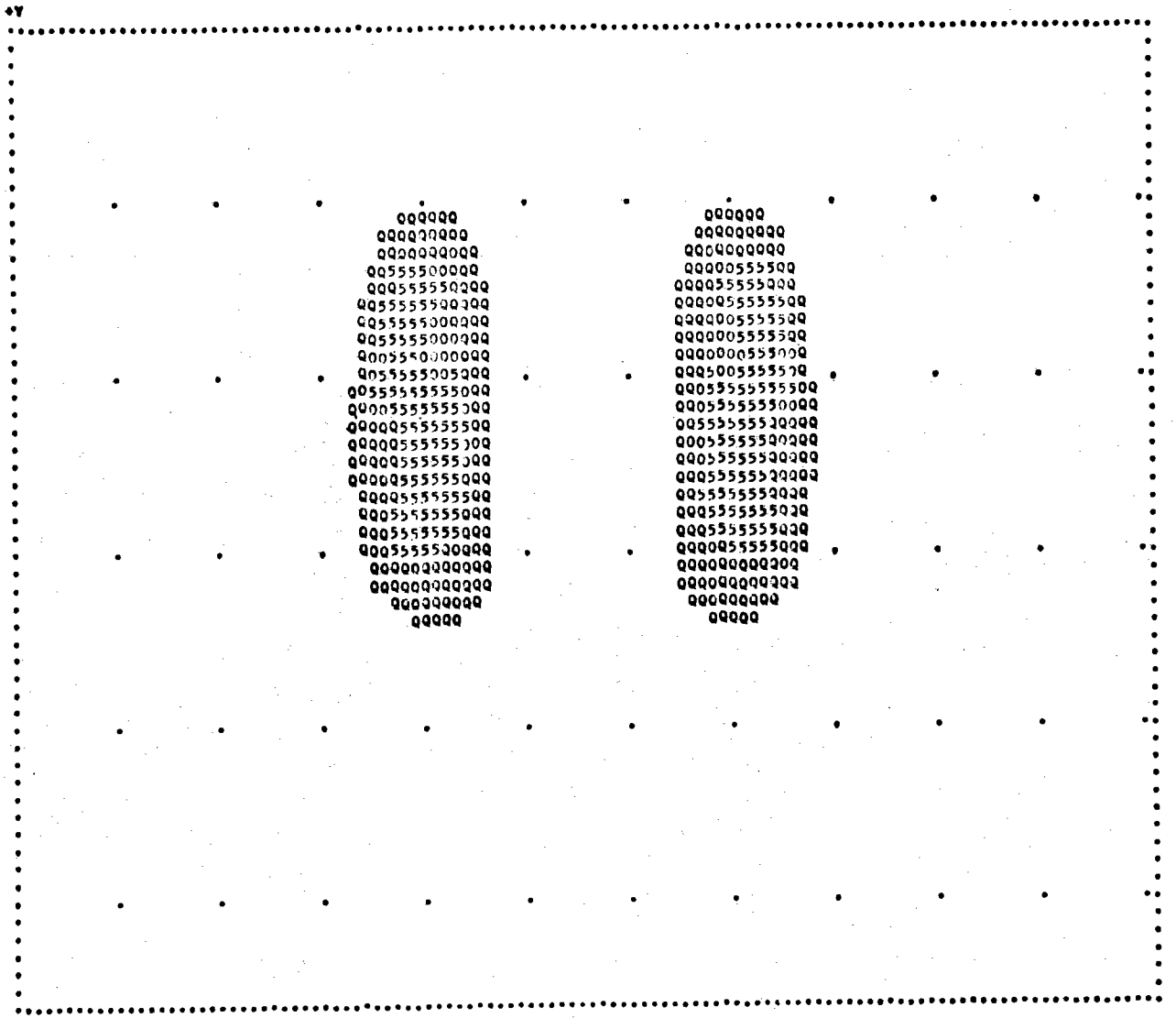

+Y
CROSS SECTION 2
+X

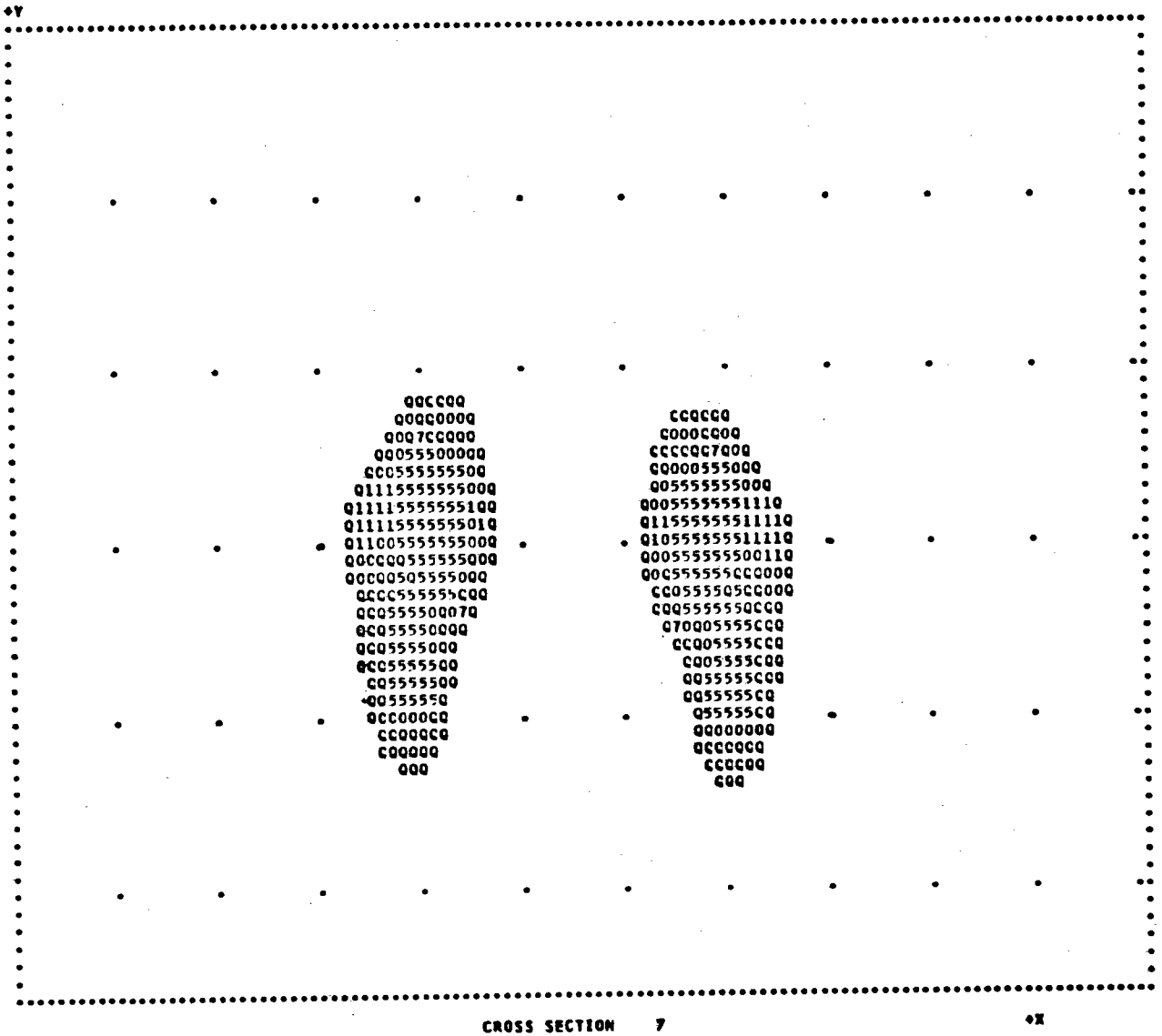
CROSS SECTION 3

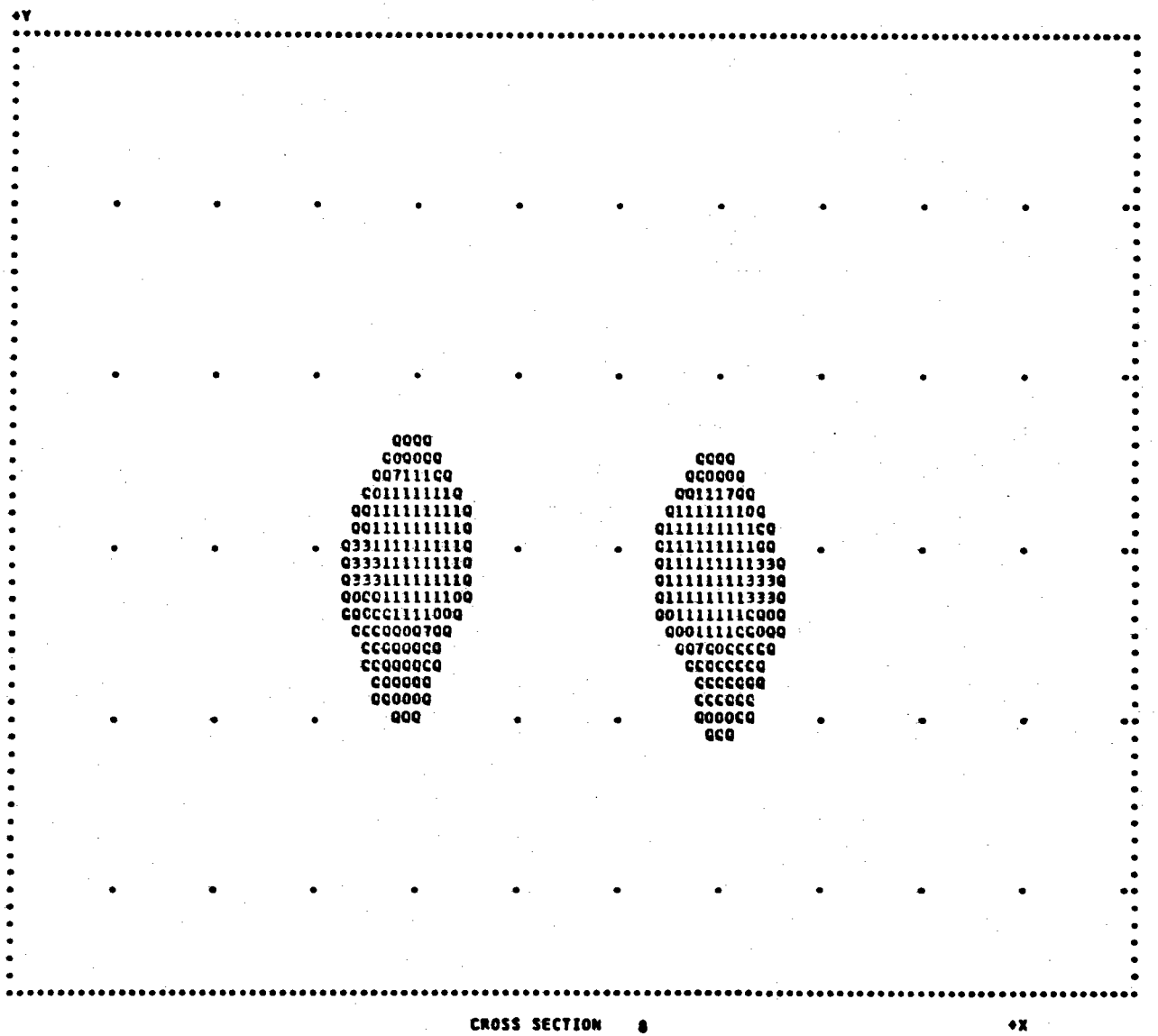


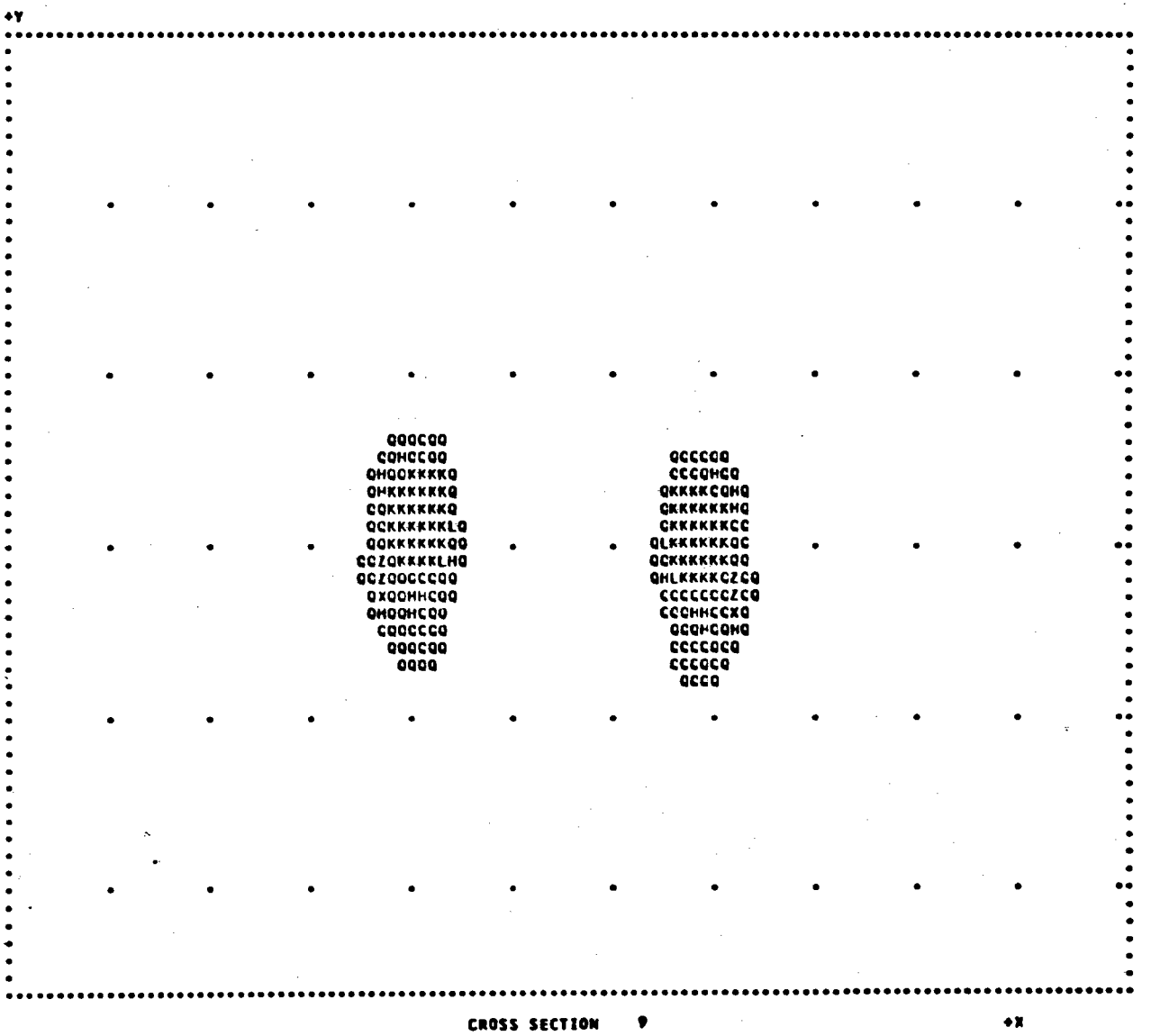
CROSS SECTION 4

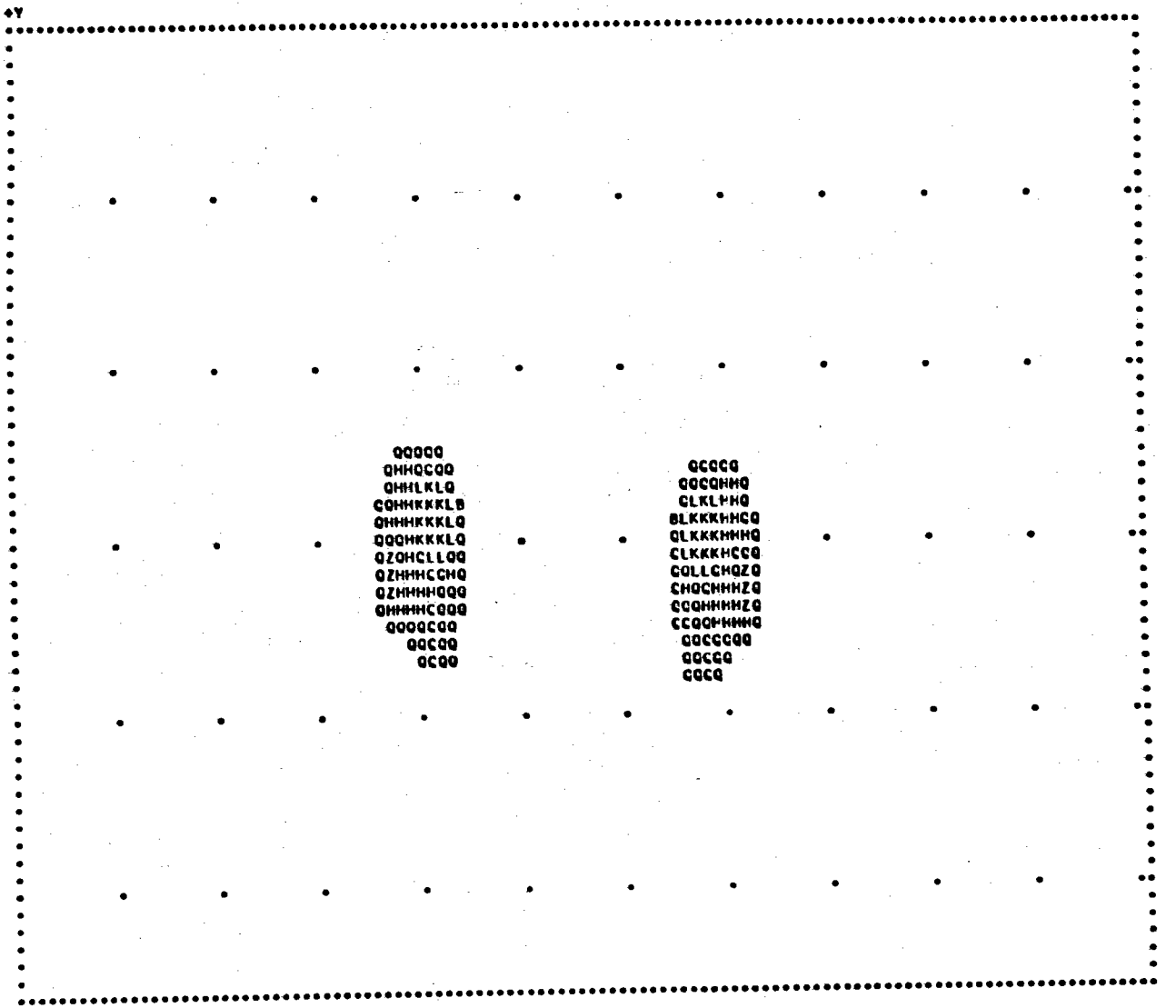

+Y
+X
CROSS SECTION 5

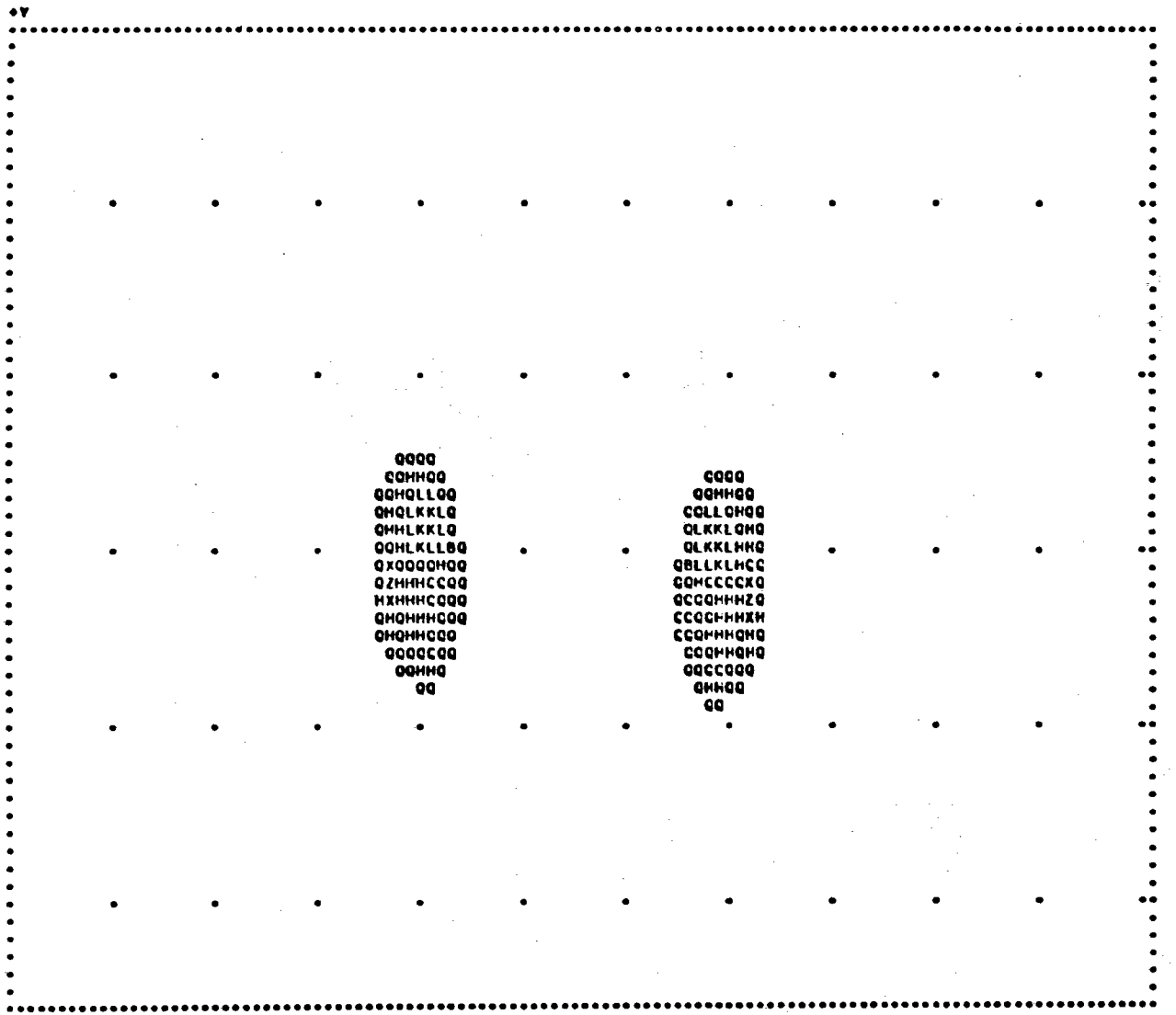

+Y
+X
CROSS SECTION 6

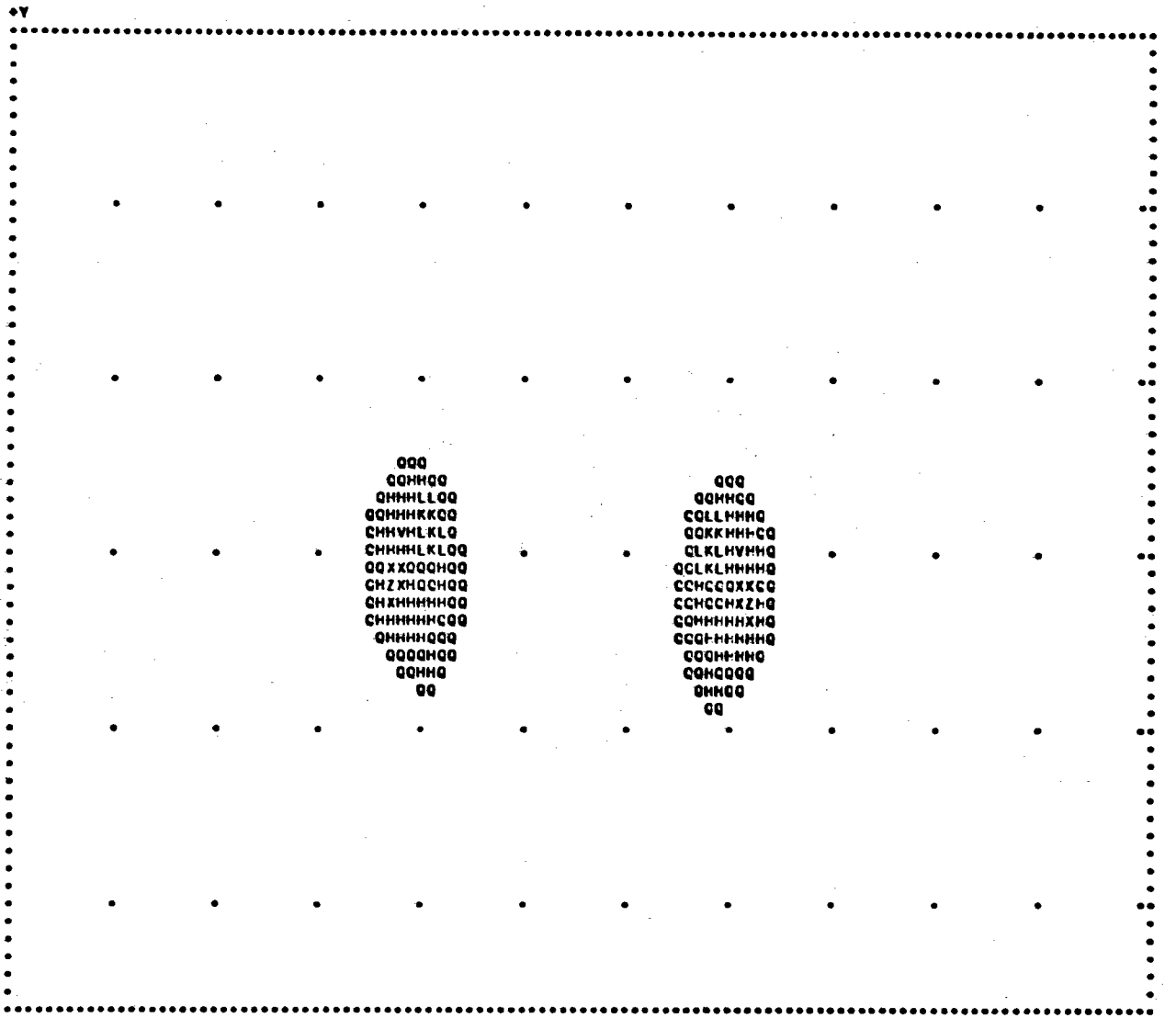

+Y
+X
CROSS SECTION 7

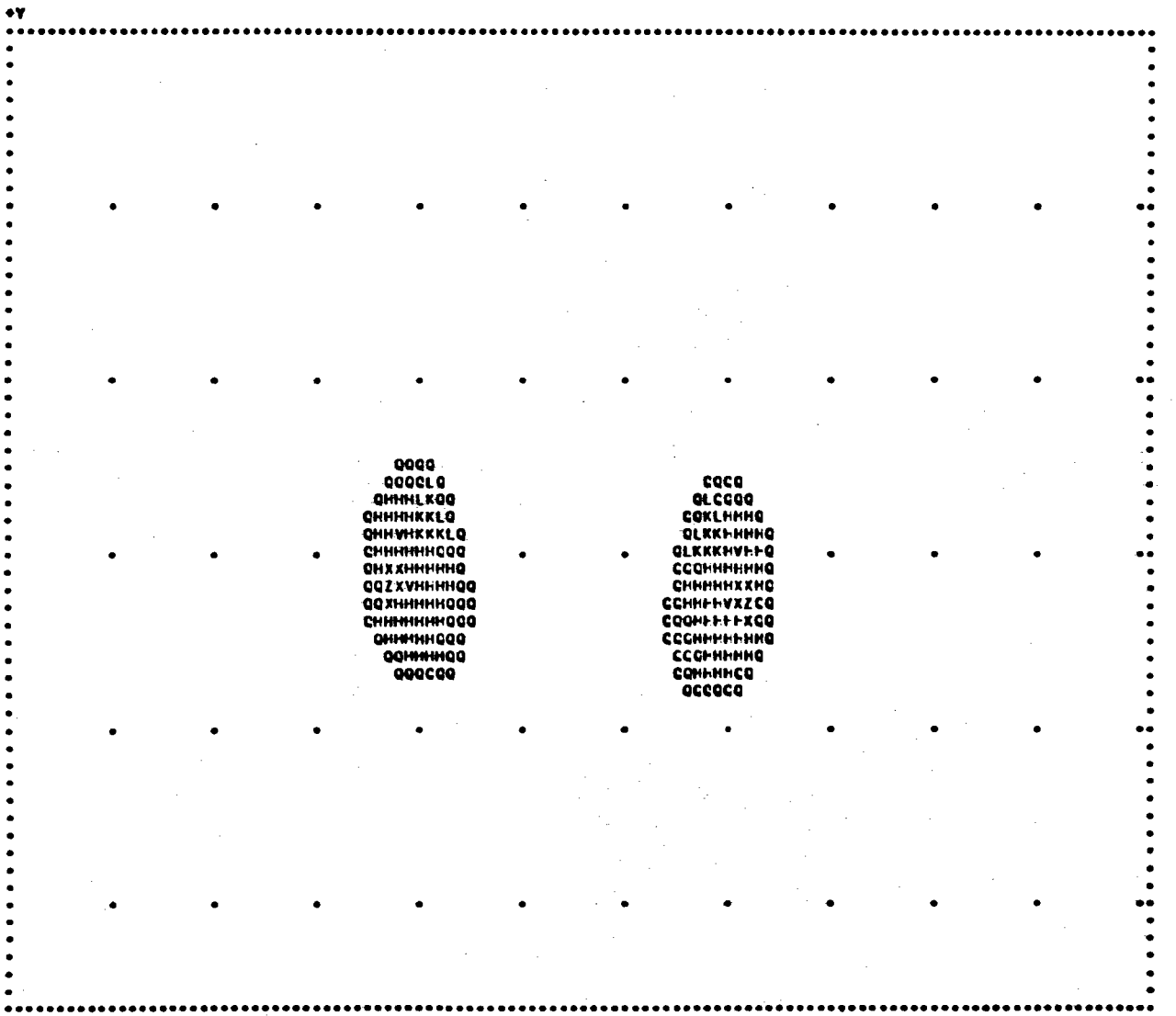

+Y
+X
CROSS SECTION 8

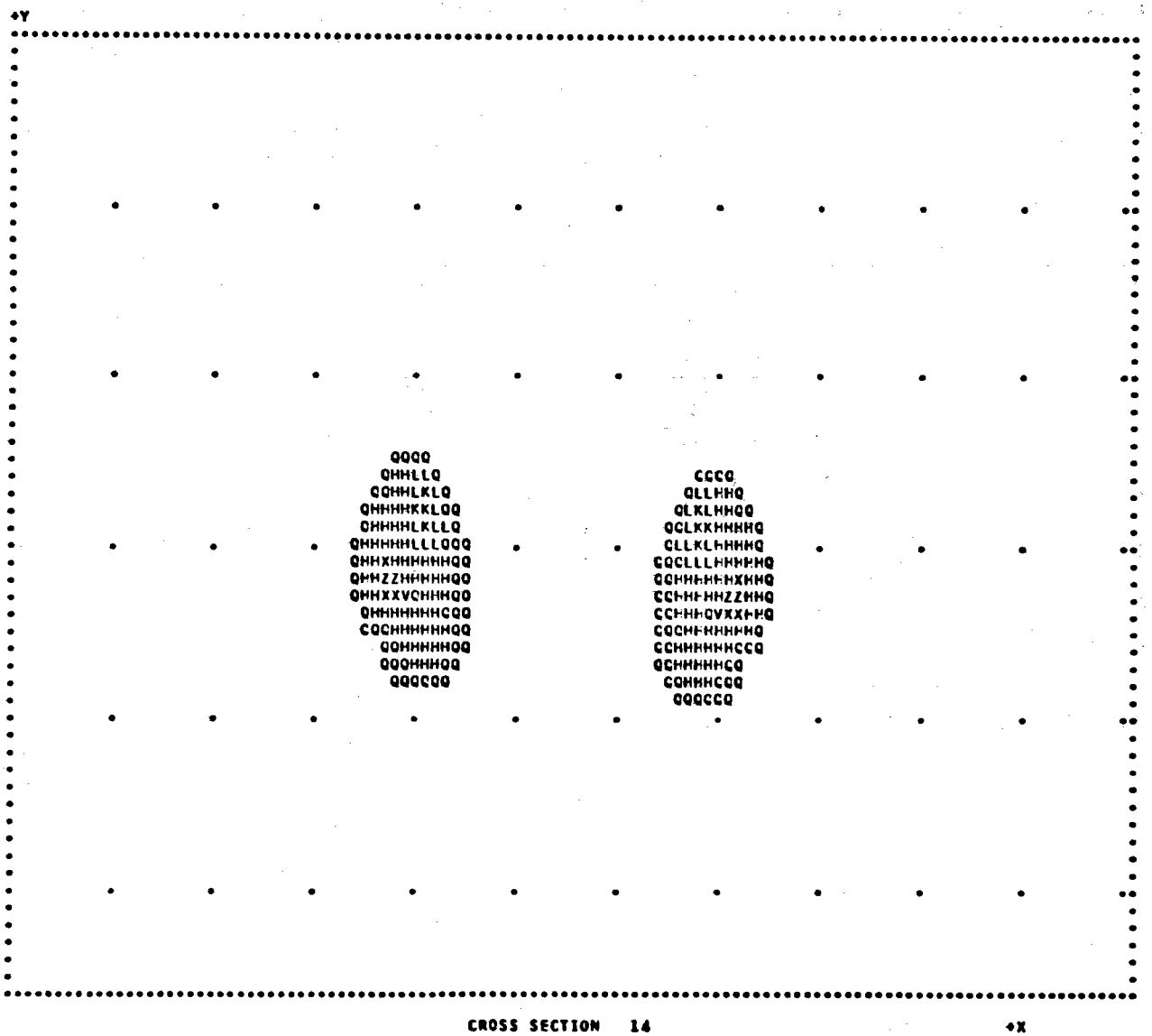

+Y
QQQCQQ
CQHCCQQ
QHQQKKKKQ
QHKKKKKKQ
CQKKKKKKQ
QCKKKKKKLQ
QQKKKKKKQQ
CCZQKKKKLHQ
QCZQQCCCQQ
QXQQHHCQQ
QHQQHCQQ
CQQCCCQ
QQQCQQ
QQQQ
QCCCQQ
CCCQHCQ
QKKKKCQHQ
CKKKKKKHQ
CKKKKKKCC
QLKKKKKKQC
QCKKKKKKQQ
QHLKKKKCZCQ
CCCCCCCZCQ
CCCHHCCXQ
QCQHCQHQ
CCCCQCQ
CCCQCQ
QCCQ
CROSS SECTION 9
+X

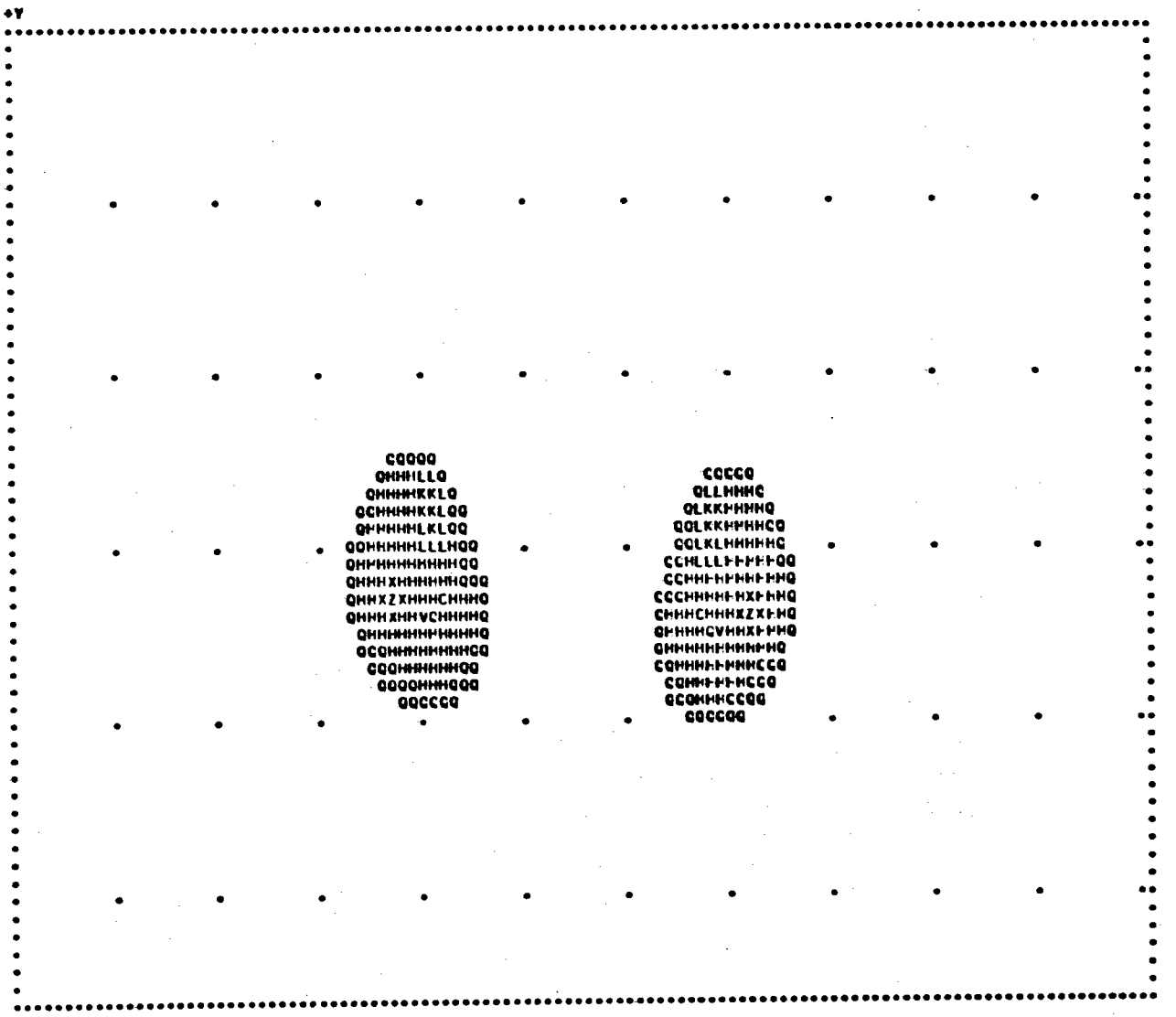

+Y
QQQQQ
QHHQCQQ
QHHLKLQ
CQHHKKKLB
QHHHKKKLQ
QQQHKKKLQ
QZQHCLLQQ
QZHHHCCHQ
QZHHHHQQQ
QHHHHCQQQ
QQQQCQQ
QQCQQ
QCQQ
QCQCQ
QQCQHHQ
QLKLPHQ
BLKKKHHCQ
QLKKKHHHQ
CLKKKHCCQ
CQLLCHQZQ
CHQCHHHZQ
CQQHHHHZQ
CCQQPHHHQ
QQCQCQQ
QQCQQ
CQCQ
CROSS SECTION 10
+X

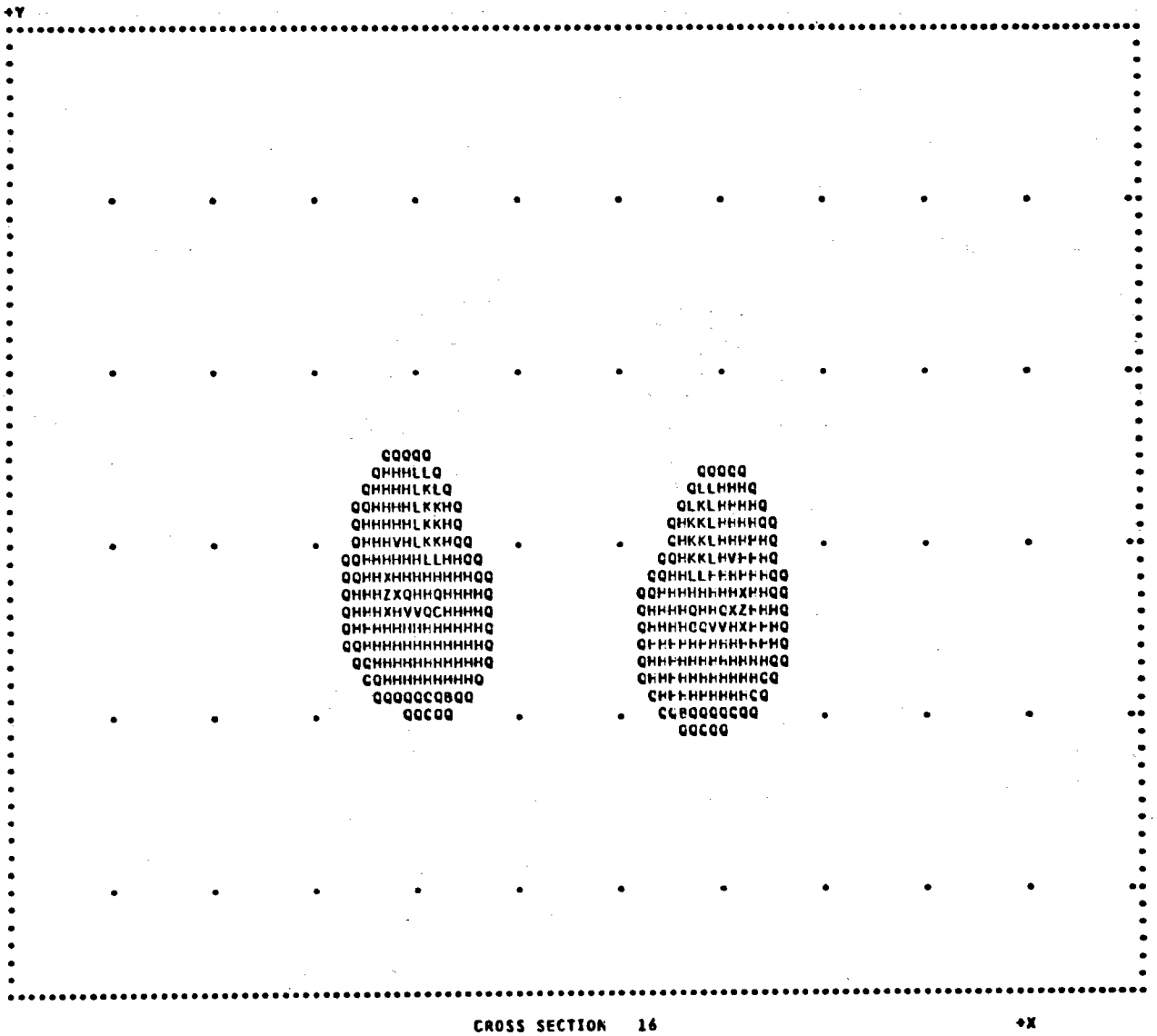

+Y
QQQQ
QQHHQQ
QQHQLLQQ
QHQLKKLQ
QHHLKKLQ
QQHLKLLBQ
QXQQQQHQQ
QZHHHCCQQ
HXHHHCQQQ
QHQHHHQQQ
QHQHHCQQ
QQQQCQQ
QQHHQ
QQ
QQQQ
QQHHQQ
QQLLQHQQ
QLKKLQHQ
QLKKLHHQ
QBLLKLHCC
QQHCCCCXQ
QCCQHHHZQ
CCQQCHHHXH
CCQHHHQHQ
CQQHHQHQ
QQCCQQQ
QHHQQ
QQ
CROSS SECTION 11
+X


+Y
QQQ
QQHHQQ
QHHHLLQQ
QQHHHKKQQ
QHHVHLKLQ
QHHHHLKLQQ
QQXXQQQHQQ
QHZXHQQHQQ
QHXHHHHHQQ
QHHHHHHCQQQ
QHHHHQQQ
QQQQHQQ
QQHHQ
QQ
QQQ
QQHHQQ
QQLLHHHQ
QQKKHHHCQ
QLKLHVHHQ
QQLKLHHHHQ
QQHQQQXXQQ
QQHQQHXZHQ
QQHHHHHXHQ
QQQHHHHHHQ
QQQHHHHHQ
QQHQQQQ
QHHQQ
QQ
CROSS SECTION 12
+X

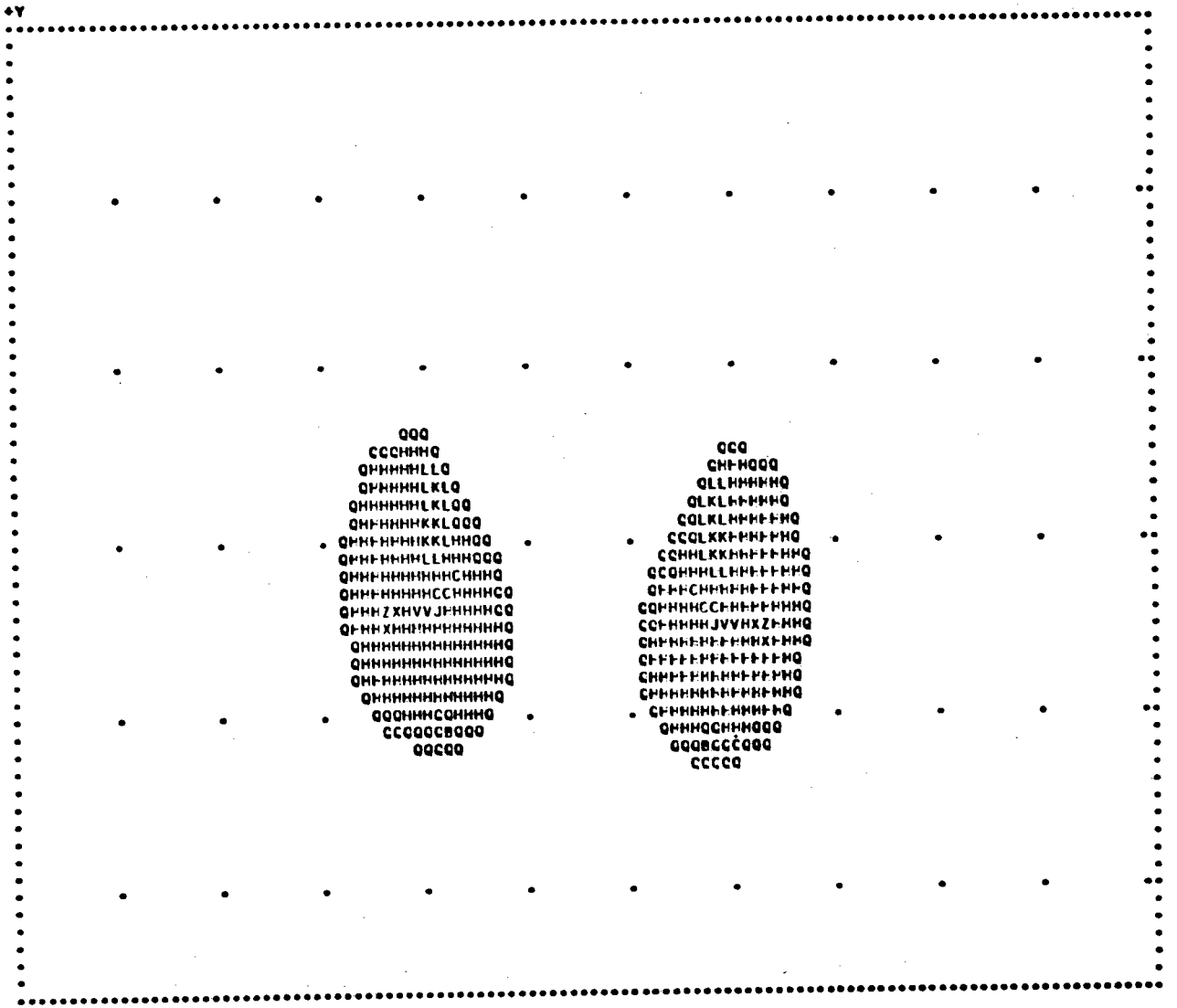

+Y
QQQQ
QQQCLQ
QHHHLKQQ
QHHHHKKLQ
QHHVHKKKLQ
CHHHHHHCQQ
QHXXHHHHHQ
QQZXVHHHHQQ
QQXHHHHHQQQ
CHHHHHHHQQQ
QHHHHHQQQ
QQHHHHQQ
QQQCQQ
CQCQ
QLCCQQ
CQKLHHHQ
QLKKHHHHQ
QLKKKHVHHQ
CCQHHHHHHQ
CHHHHHXXHQ
CCHHHHHVXZCQ
CQQHHHHHXCQQ
CCCHHHHHHHQ
CCCHHHHHQ
CQHHHHCQ
QCCQCQ
CROSS SECTION 13
+X


+Y
QQQQ
QHHLLQ
QQHHLKLQ
QHHHHKKLQQ
QHHHHLKLLQ
QHHHHHLLLQQQ
QHHXHHHHHHQQ
QHHZZHHHHHQQ
QHHXXVQHHHQQ
QHHHHHHHCQQ
CQCHHHHHHQQ
QQHHHHHQQ
QQQHHHQQ
QQQCQQ
CCCQ
QLLHHQ
QLKLHHQQ
QQLKKHHHHQ
QLLKLHHHHQ
CQQLLLHHHHHQ
QQHHHHHHXHHQ
CCHHHHHHZZHHQ
CCHHHQVXXHHQ
CQCHHHHHHHQ
CCHHHHHHCCQ
QCHHHHHCQ
CQHHHCQQ
QQQCCQ
CROSS SECTION 14
+X

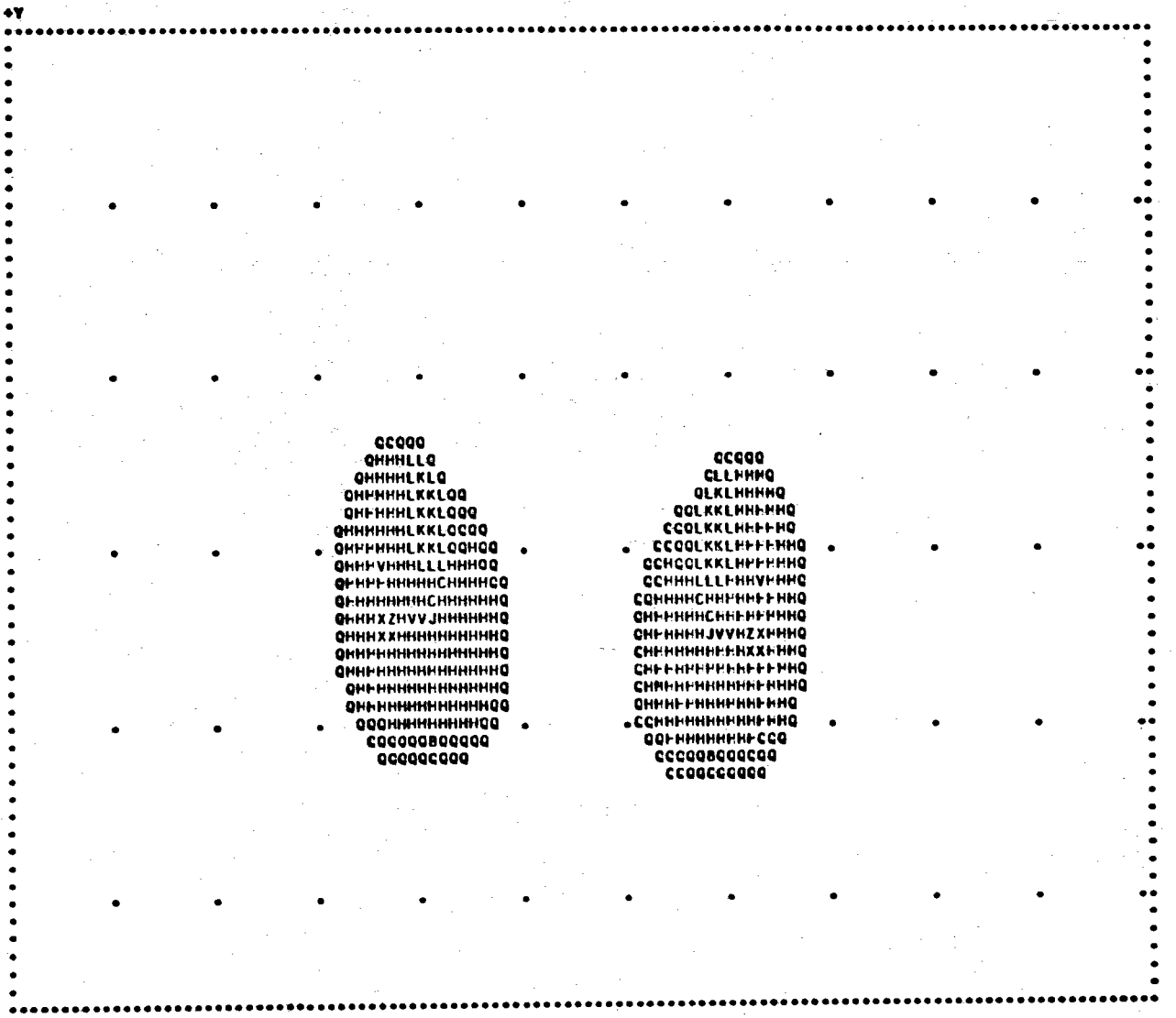

+Y
+X
CROSS SECTION 15



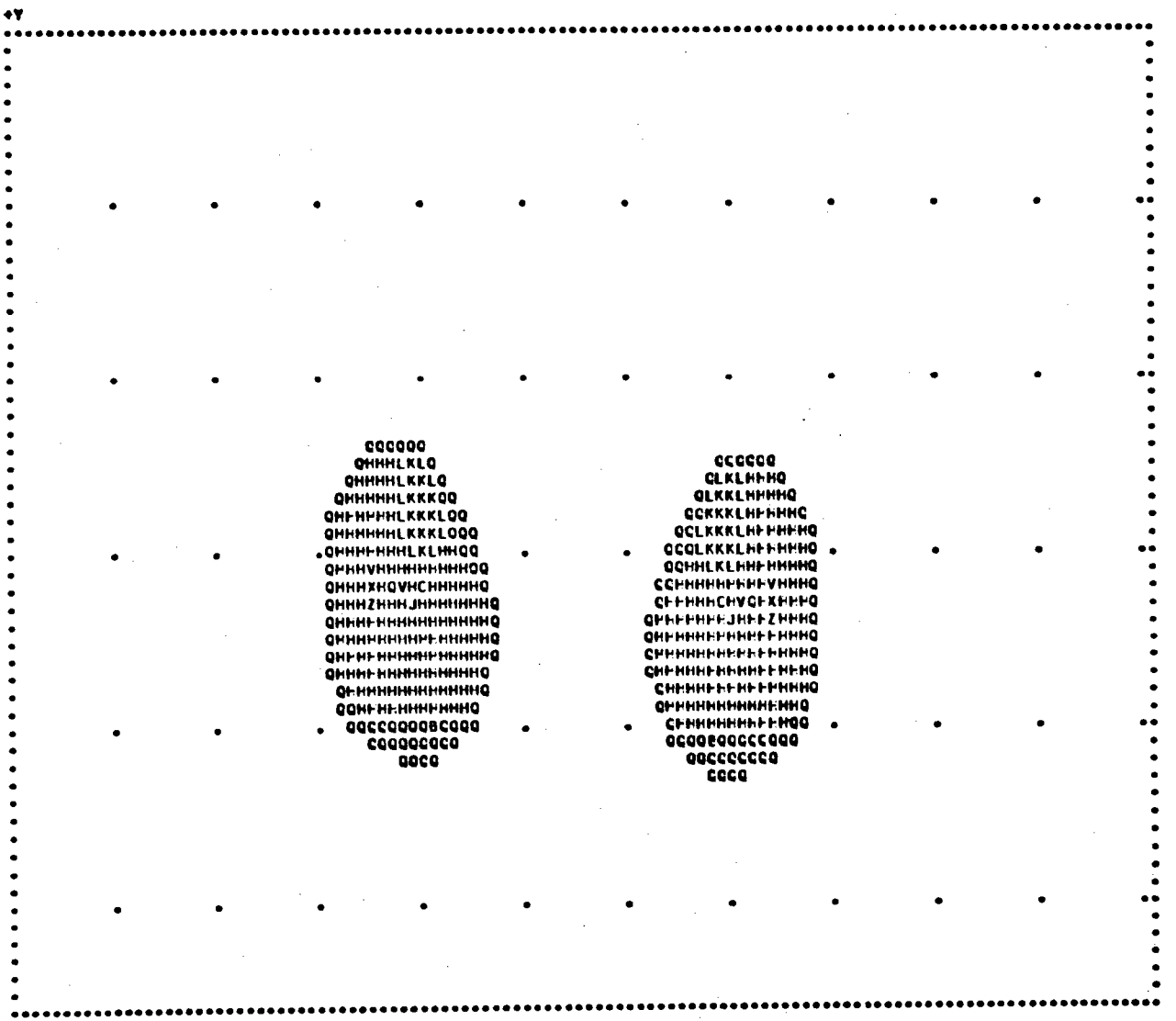
CROSS SECTION 16



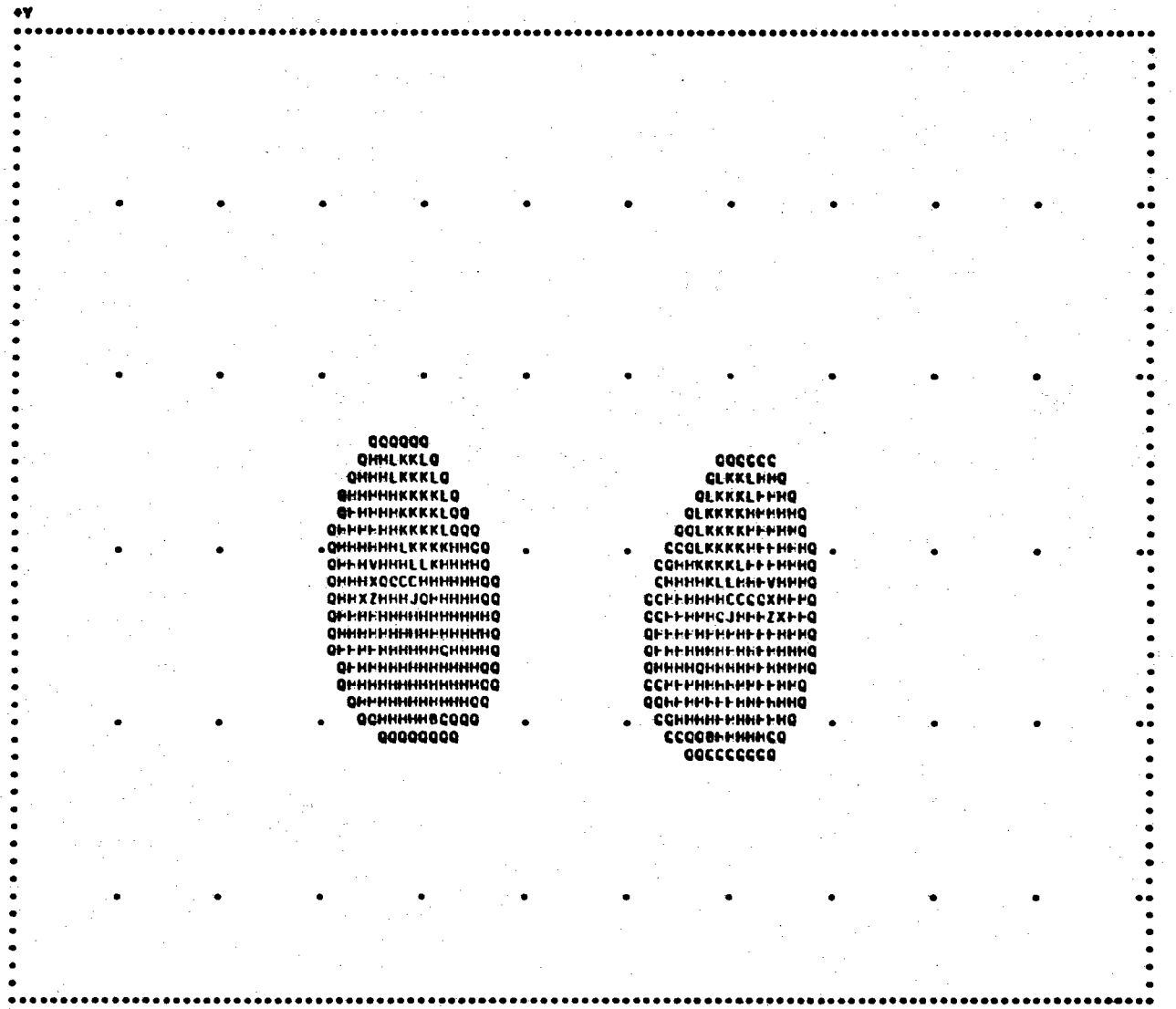
CROSS SECTION 17

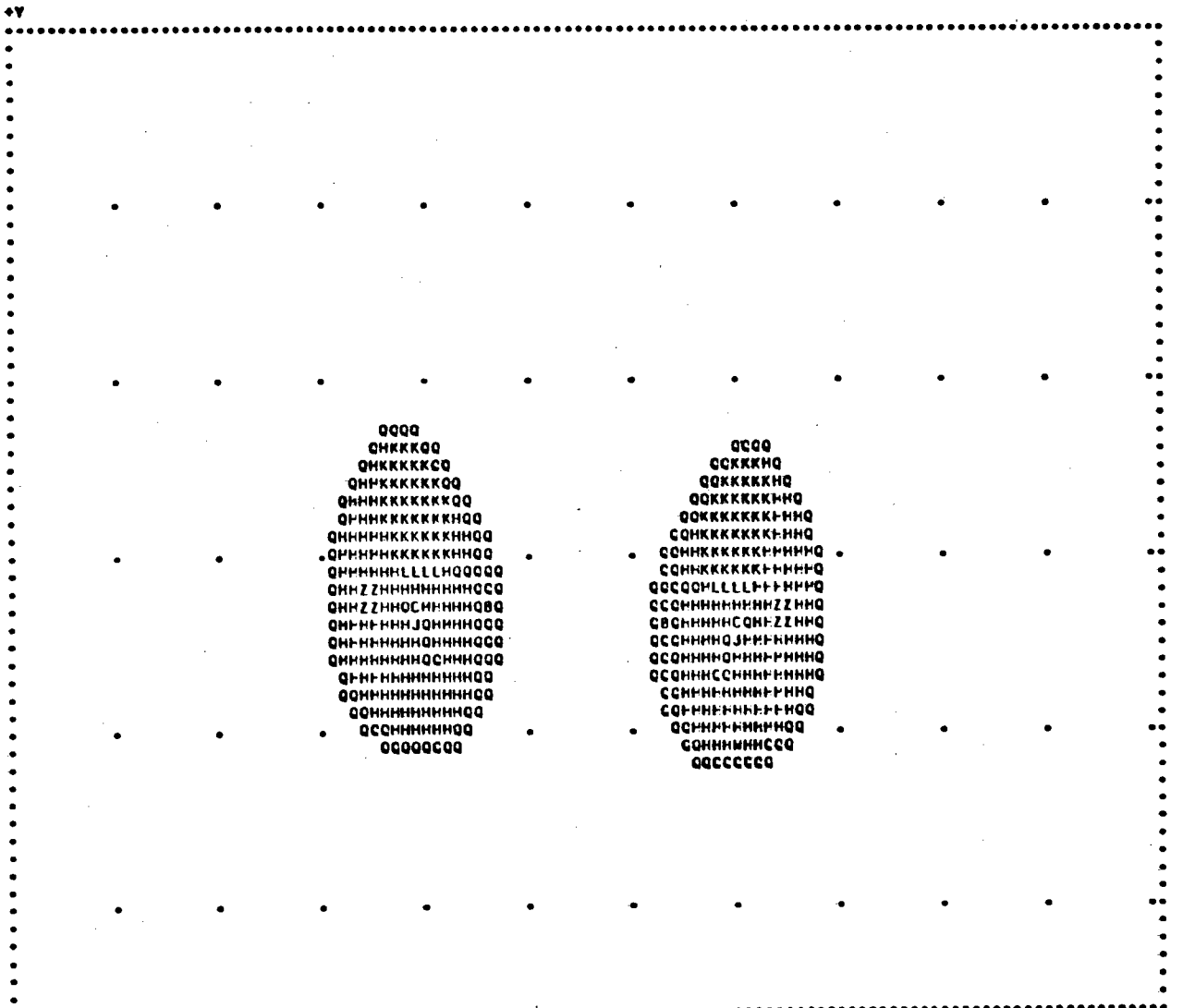


CROSS SECTION 18

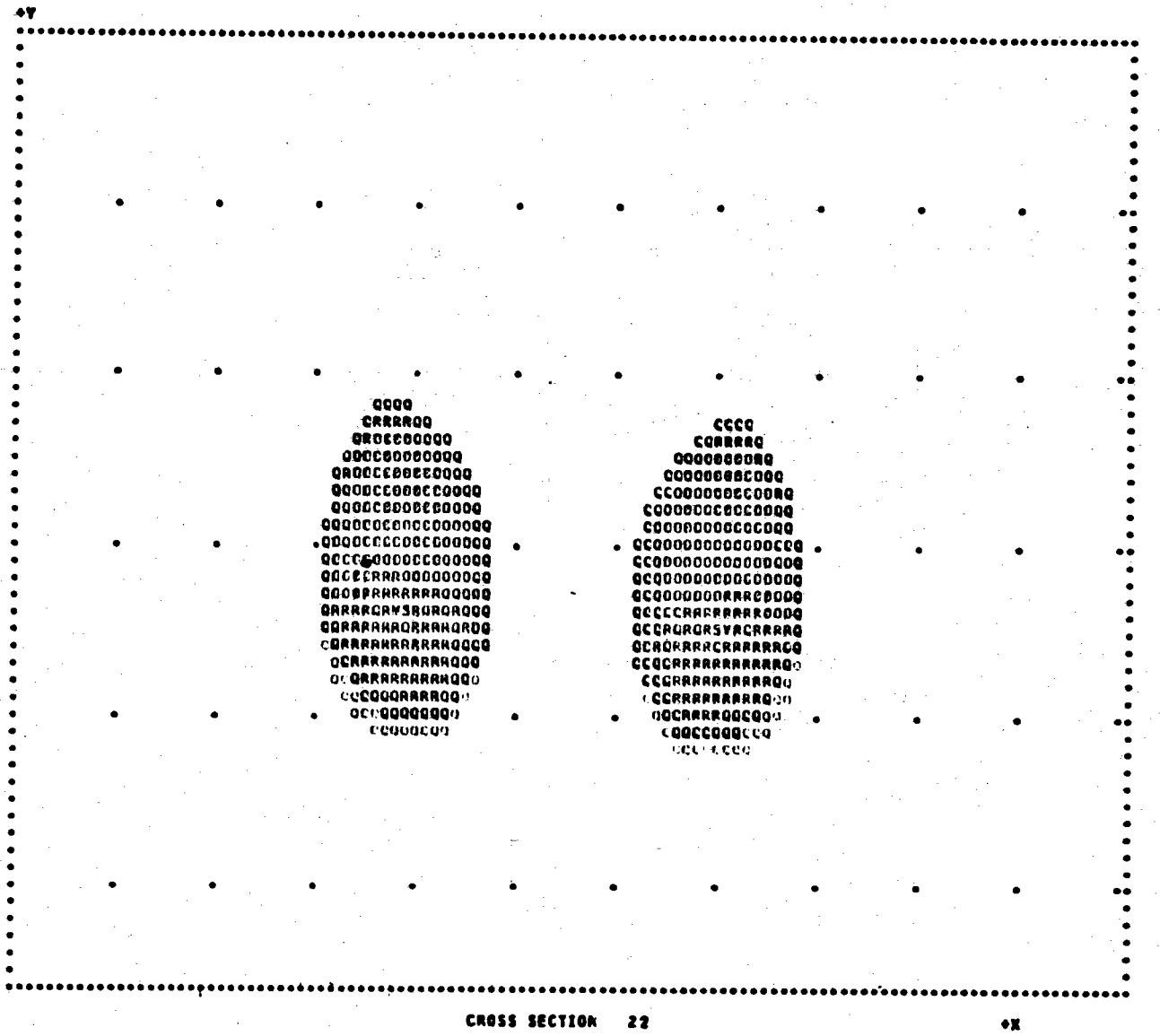


CROSS SECTION 19

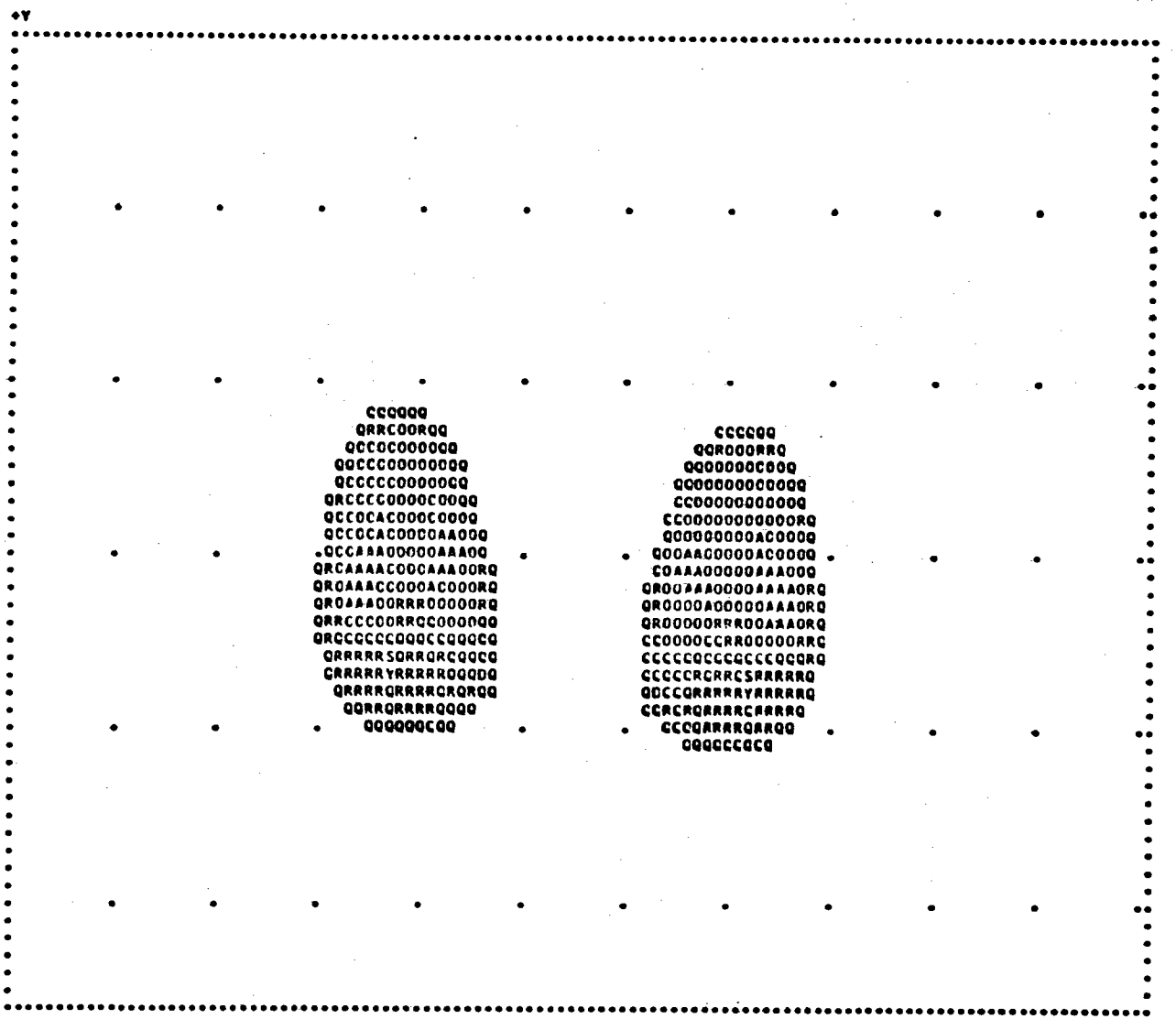

+Y
+X
CROSS SECTION 20

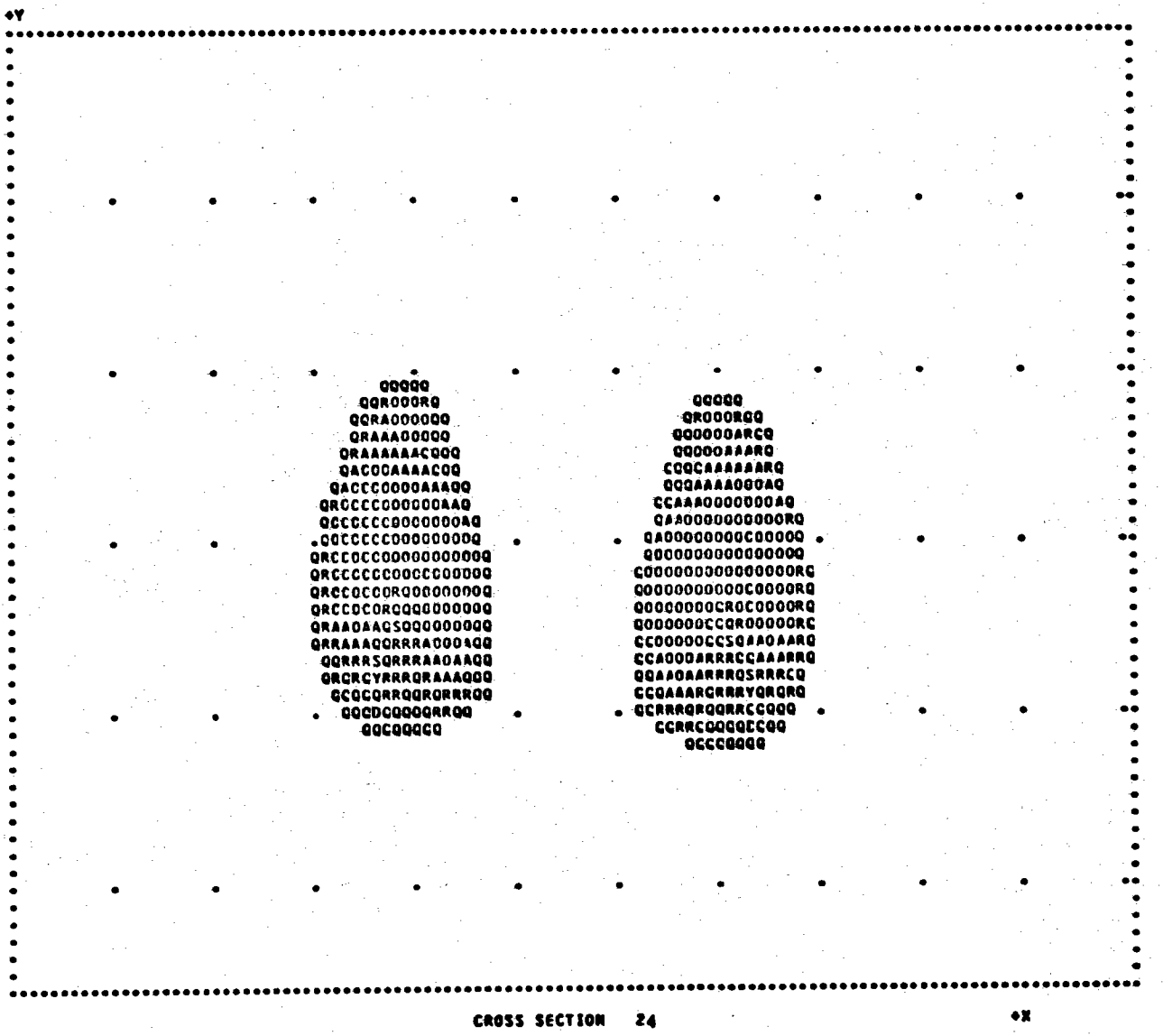


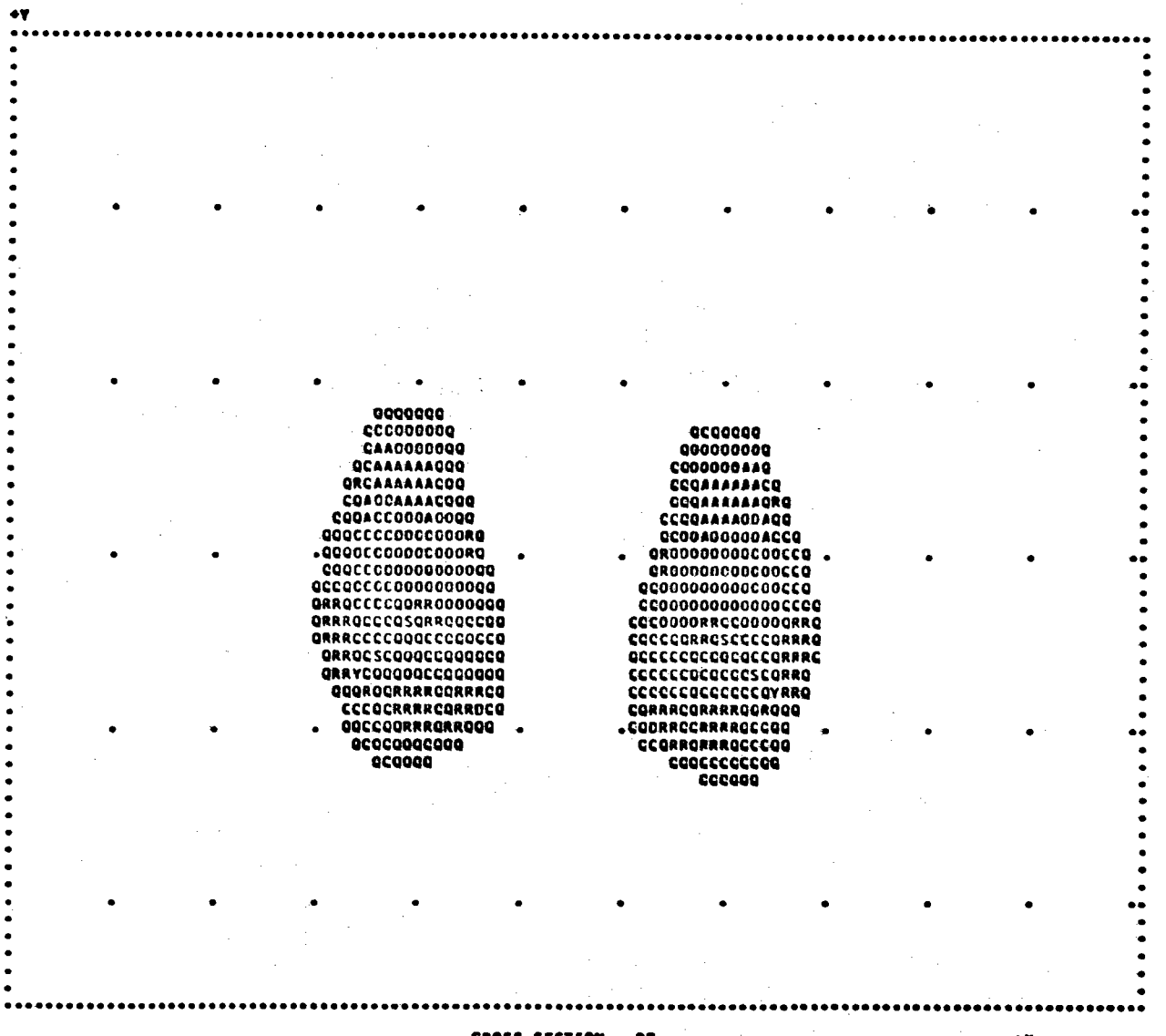
CROSS SECTION 21



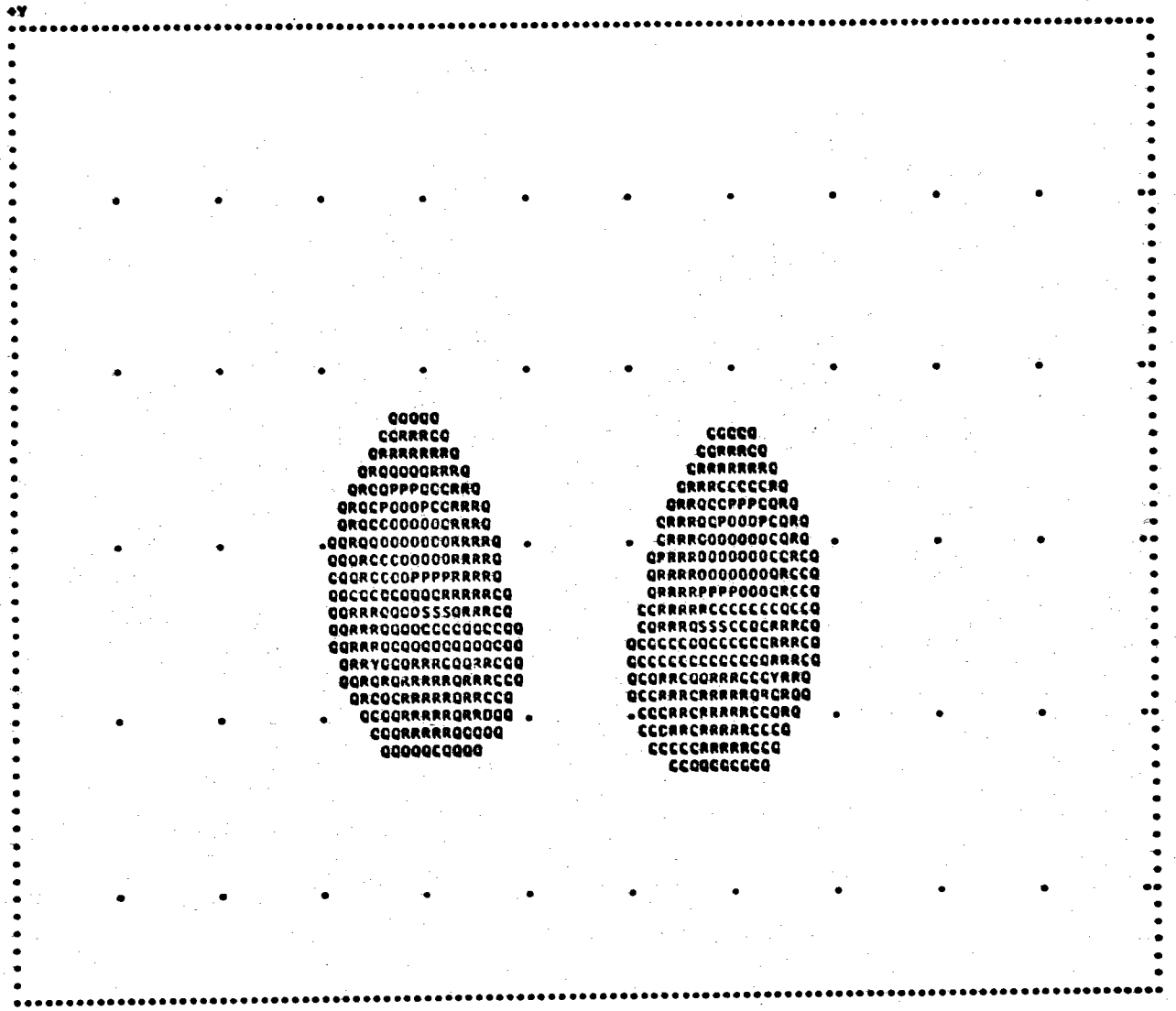
CROSS SECTION 22

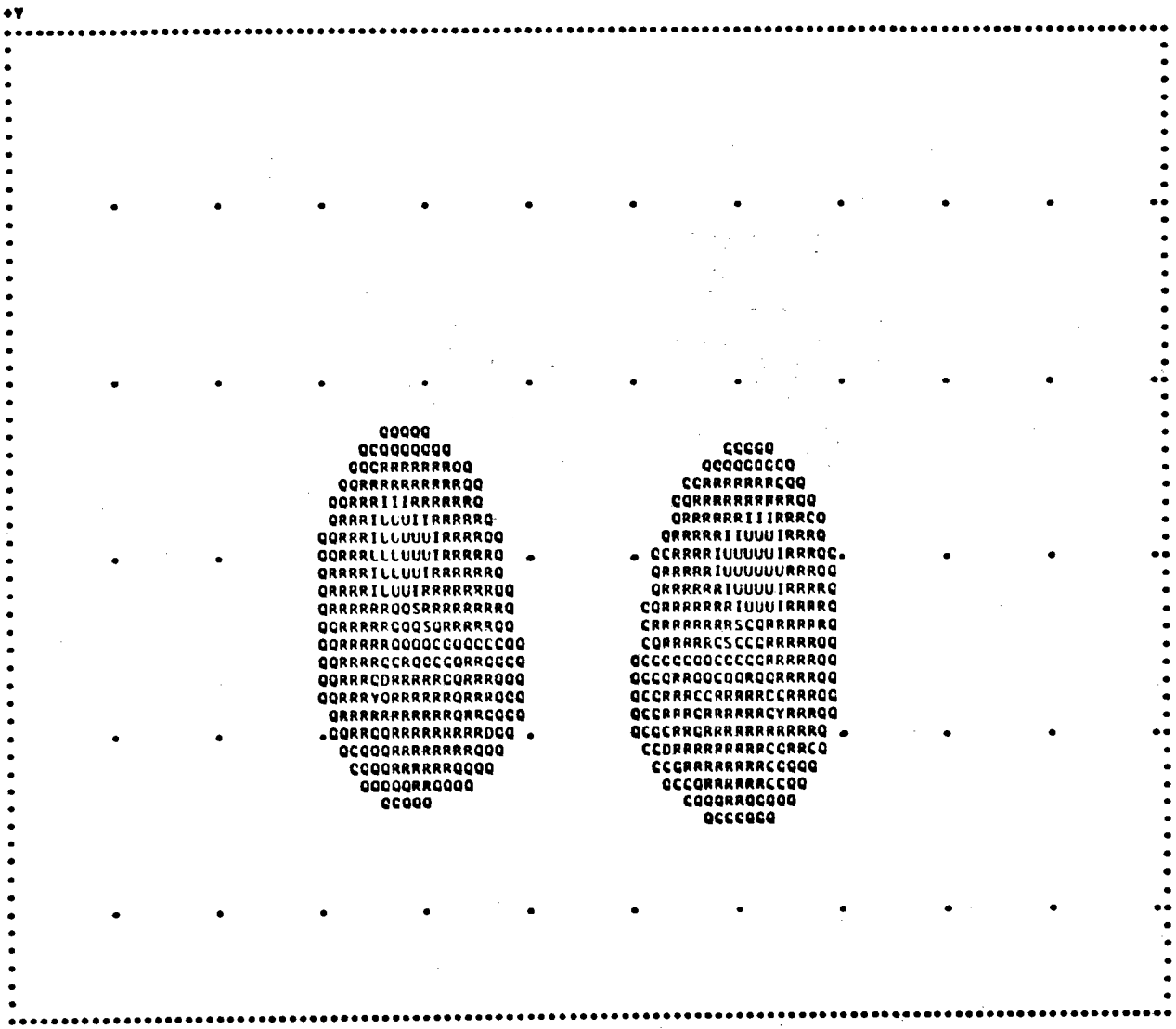

+Y
CCQQQQ
QRRCOORQQ
QCCOCOOOOQQ
QQCCCOOOOOOQQ
QCCCCCOOOOOCQ
QRCCCCOOOOCOOQQ
QCCOCACOOOCOOOQ
QCCOCACOOCOAAOOQ
QCCAAAOOOOOAAAOQ
QRCAAAACOOCAAAOORQ
QRCAAACCOOOACOOORQ
QRCAAAOORRROOOOORQ
QRRCCCOORRQCOOOOQQ
QRQCQCCCQQQCCQQQCQ
QRRRRRSQRRQRCQQCQ
QRRRRRYRRRRRQQQDQ
QRRRRQRRRRCRQRQQ
QQRRQRRRRQQQQ
QQQQQQCQQ
CCCCQQ
QQRQOORRQ
QQQOOOOCOOQ
QQOOOOOOOOOQQ
CCOOOOOOOOOOQ
CCOOOOOOOOOORQ
QOOOOOOOOACOOOQ
QOOAAOOOOOACOOOQ
COAAAOOOOOAAAOOQ
QROOAAAOOOOAAAAORQ
QROOOOAOOOOOAAAORQ
QROOOOORRROOAAAORQ
CCOOOOCCRROOOOORRQ
CCCCCQCCCQCCCCQCQRQ
CCCCCRCRRCSRRRRRQ
QDCCQRRRRRYRRRRRQ
CCRCRQRRRRCRRRRQ
CCCQRRRRQRRQQ
QQQQCCQCQ
CROSS SECTION 23
+X

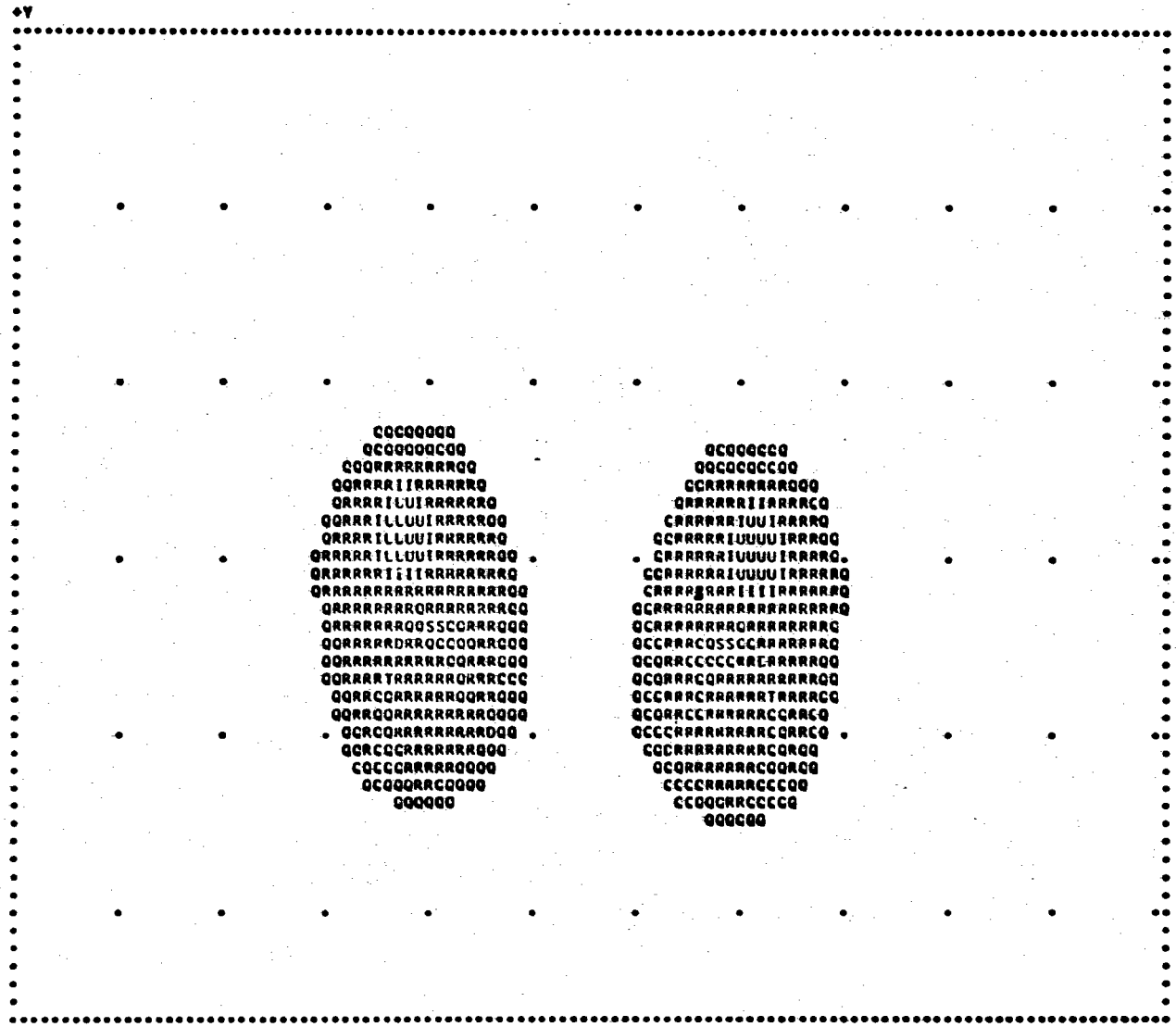

+Y
+X
CROSS SECTION 24



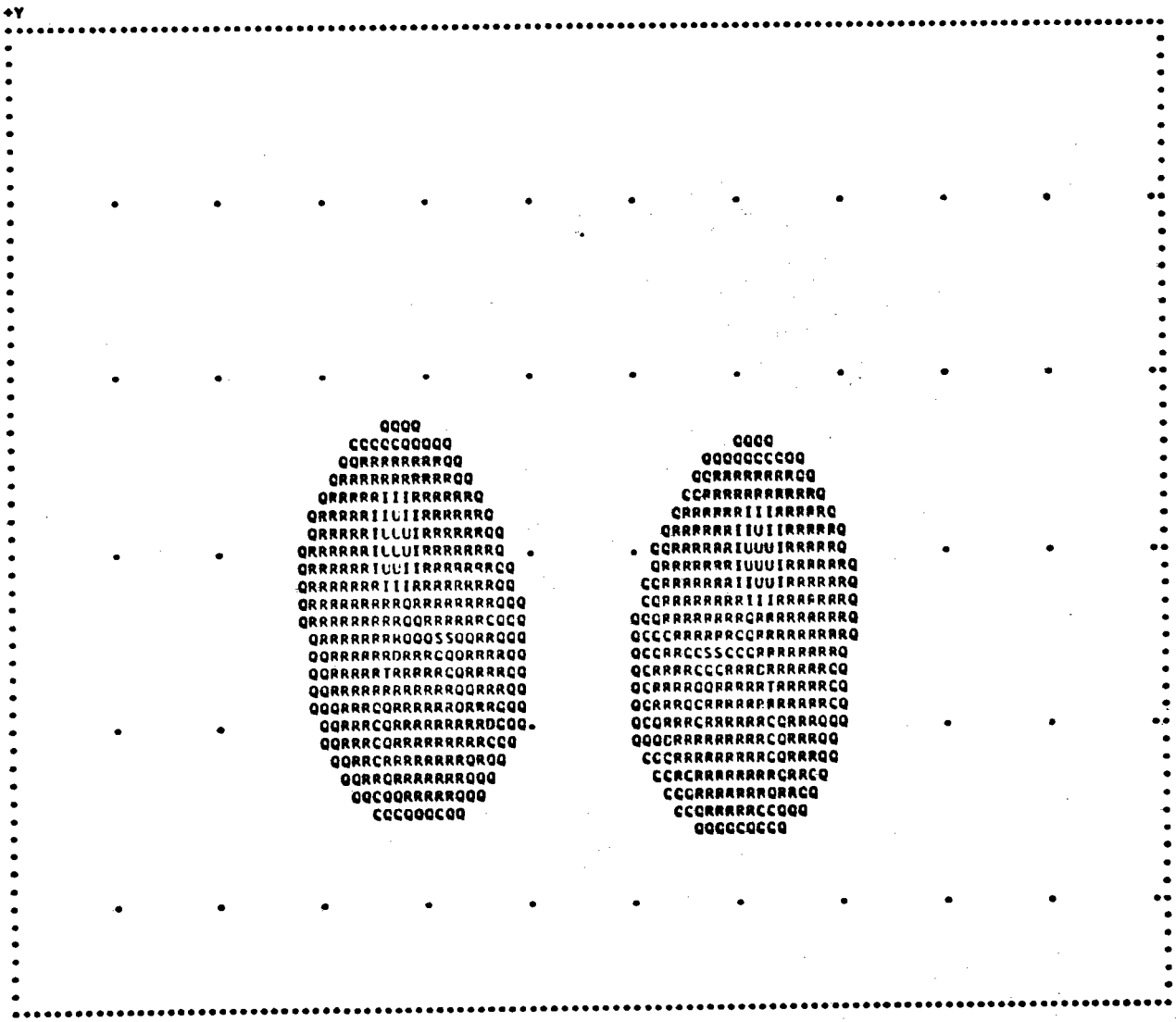
CROSS SECTION 25

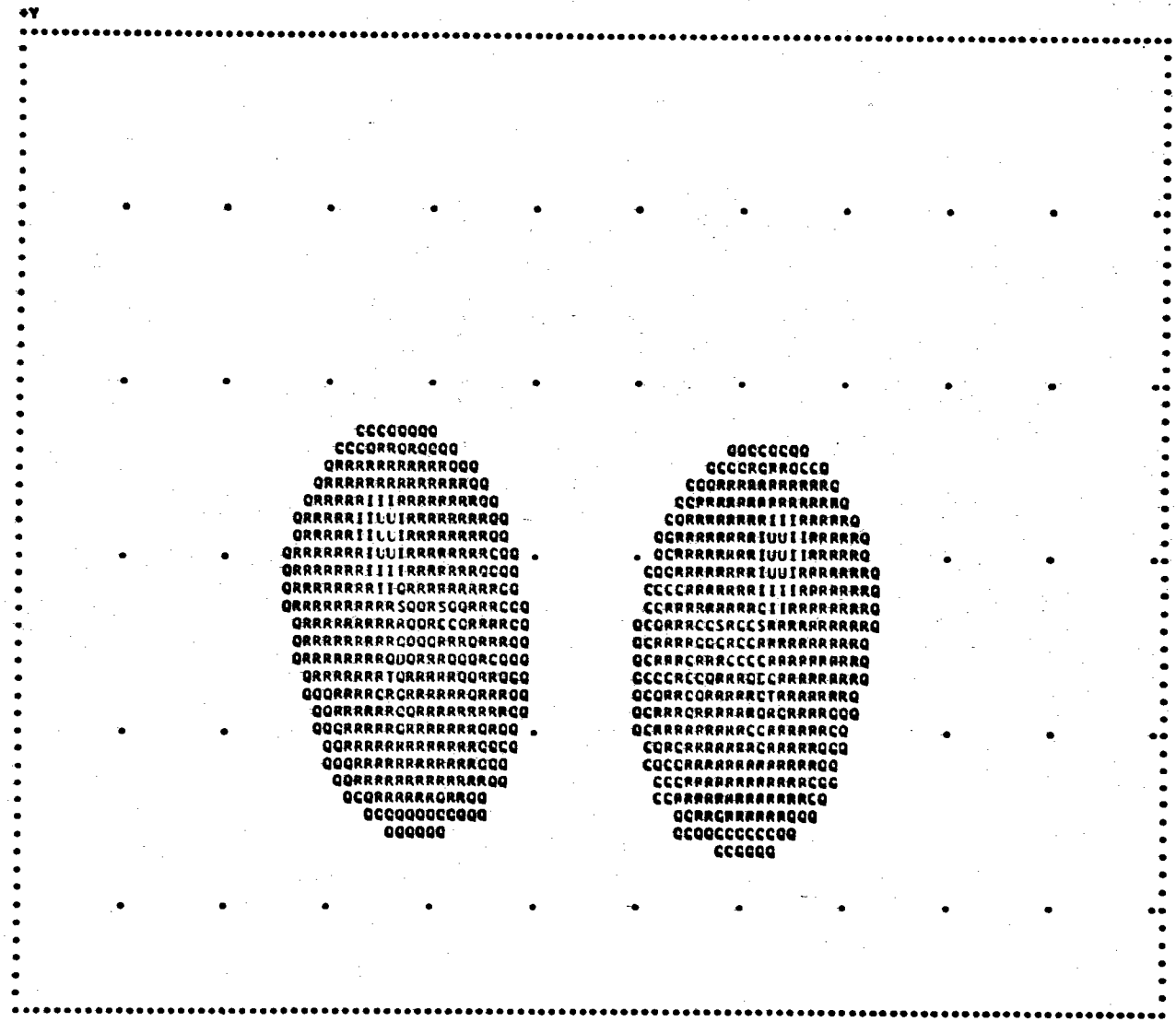

+Y
+X
CROSS SECTION 26


Y
X
CROSS SECTION 27

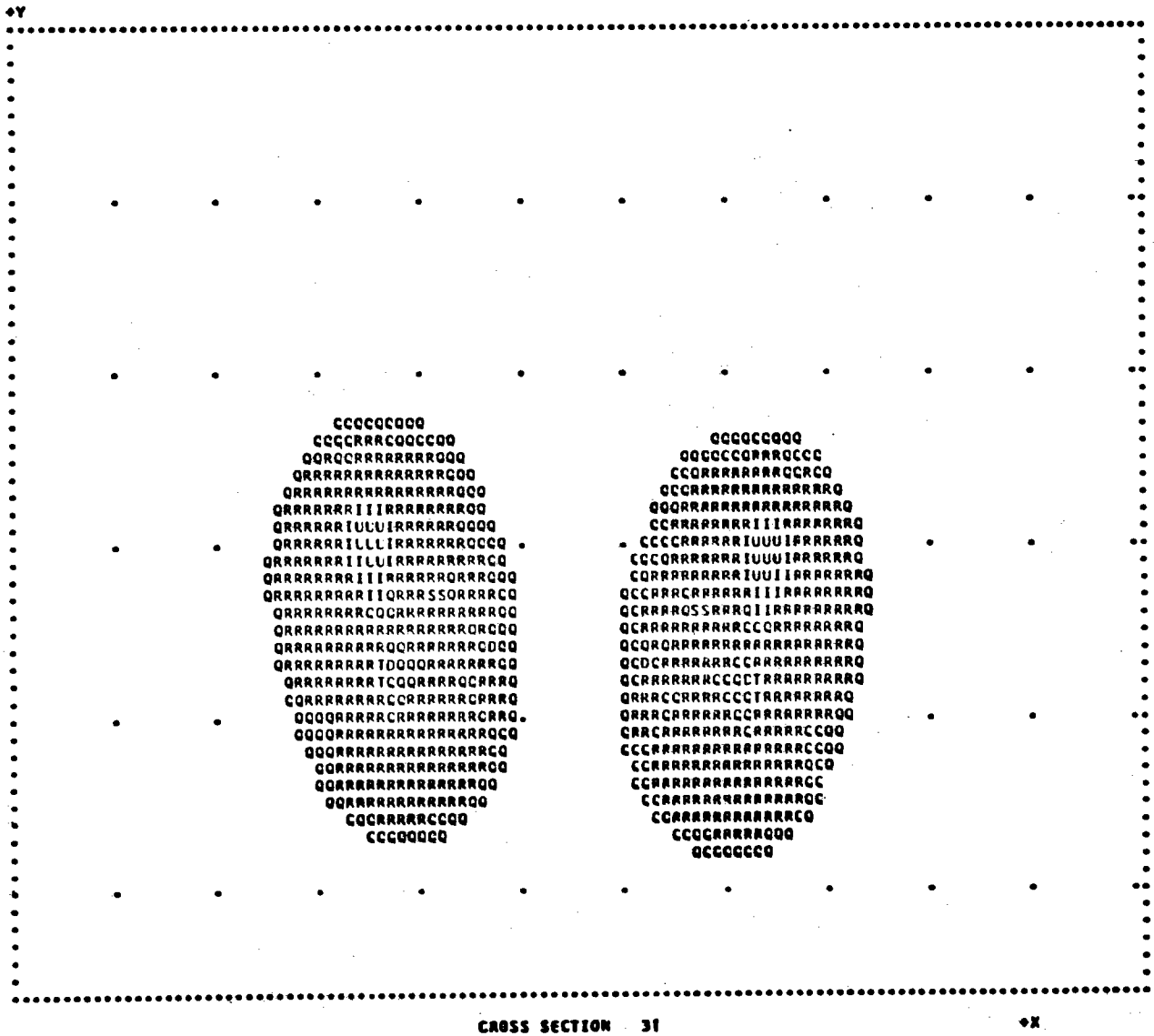


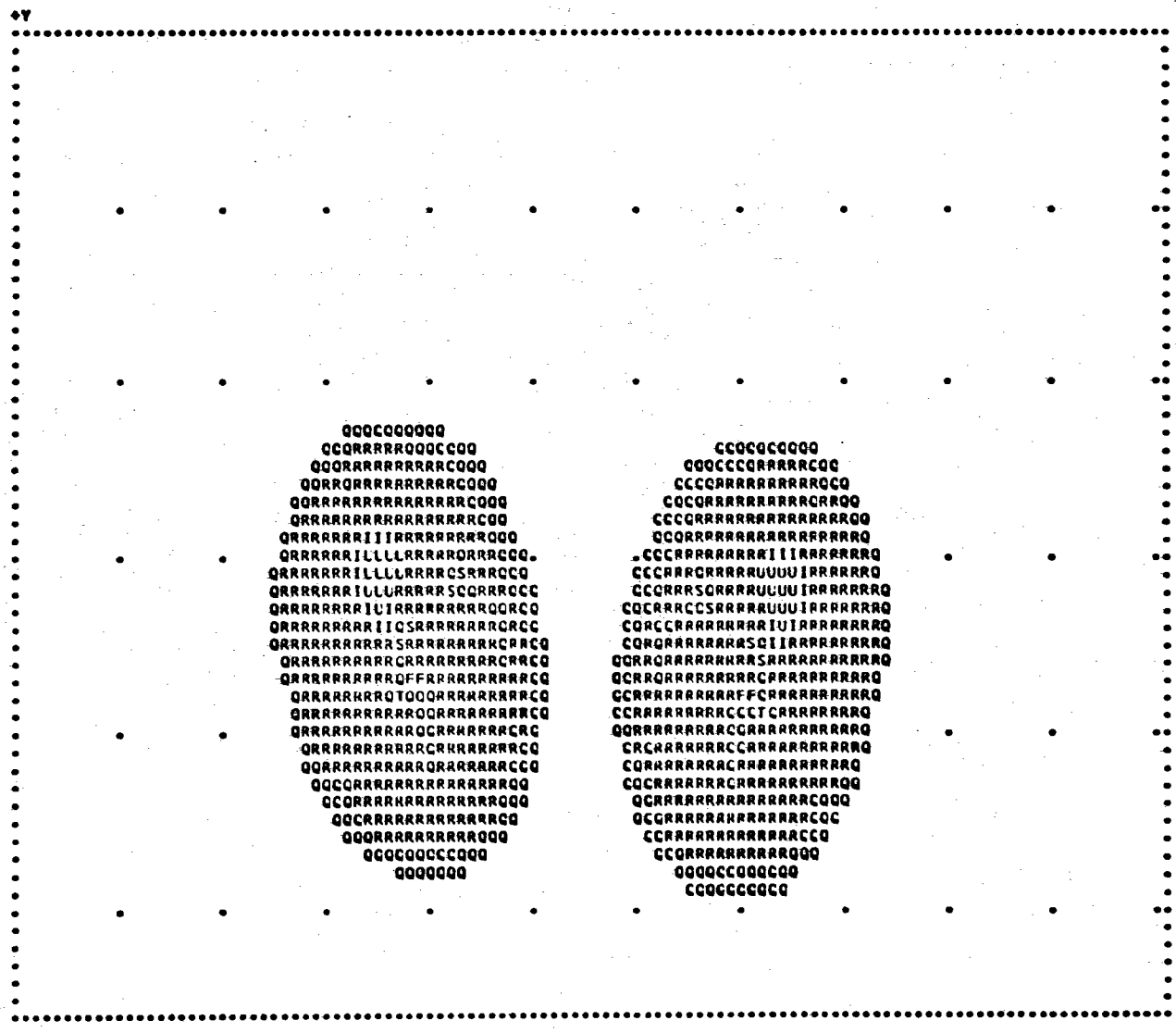
CROSS SECTION 20

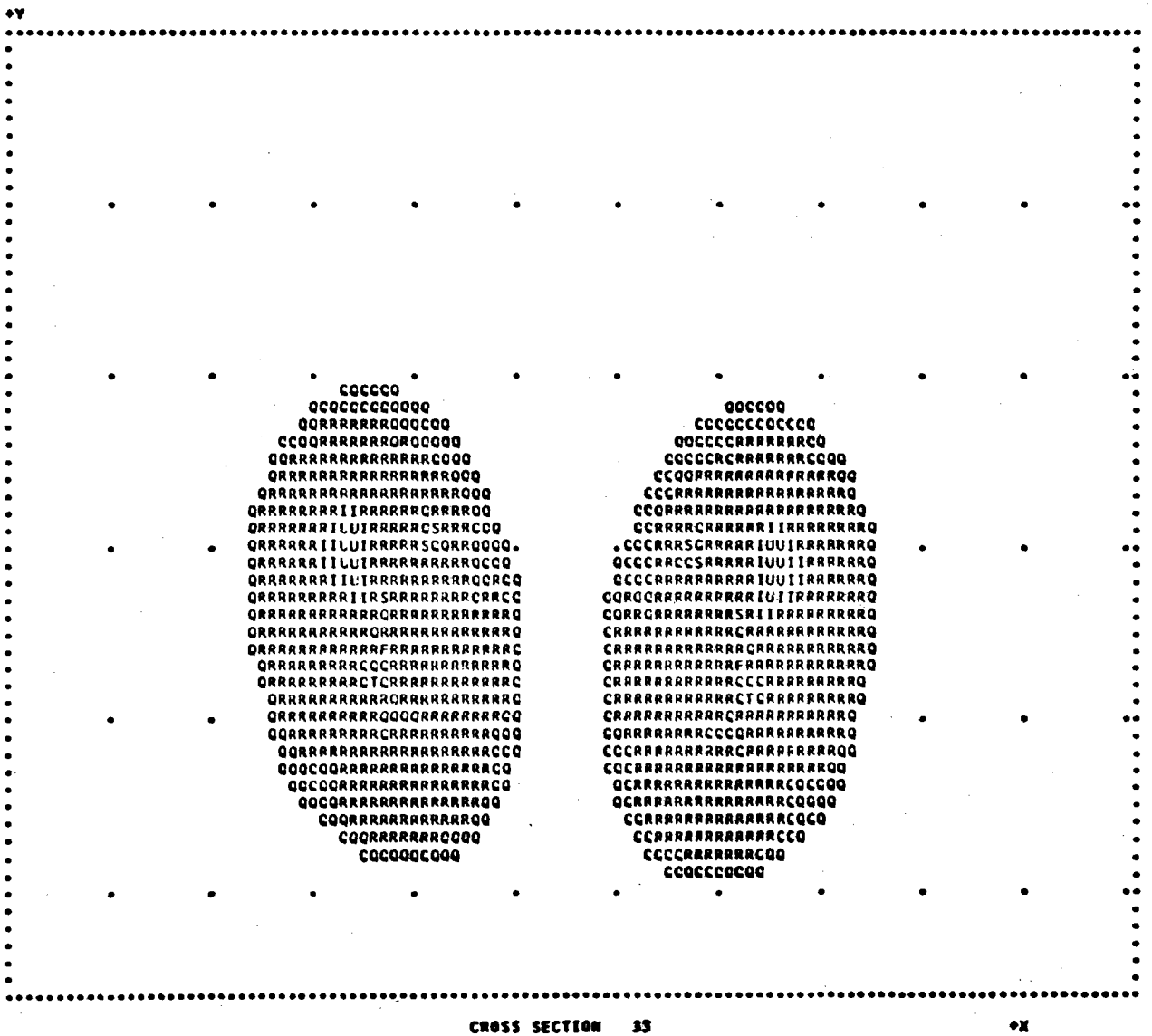

+Y
+X
CROSS SECTION 39

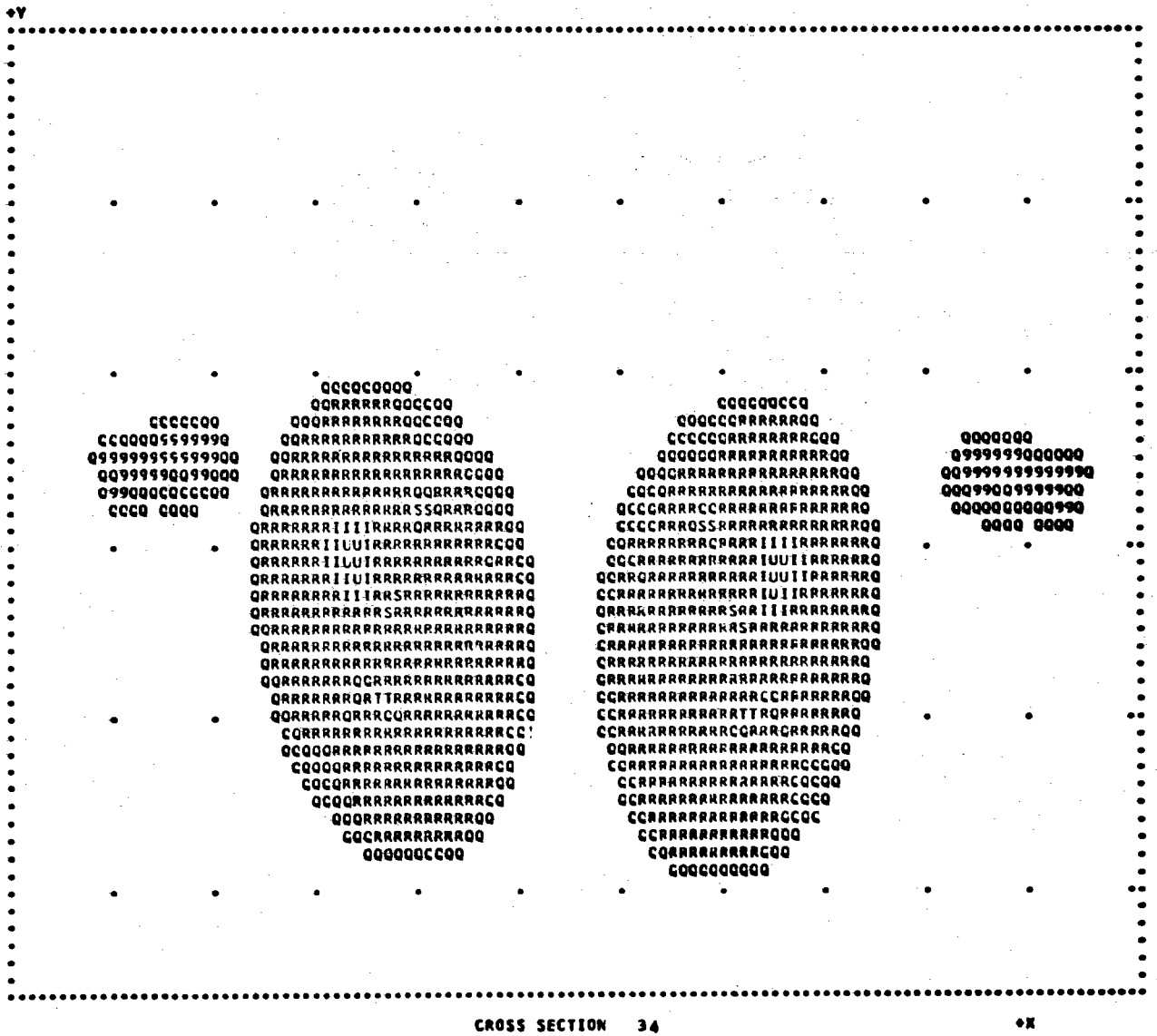


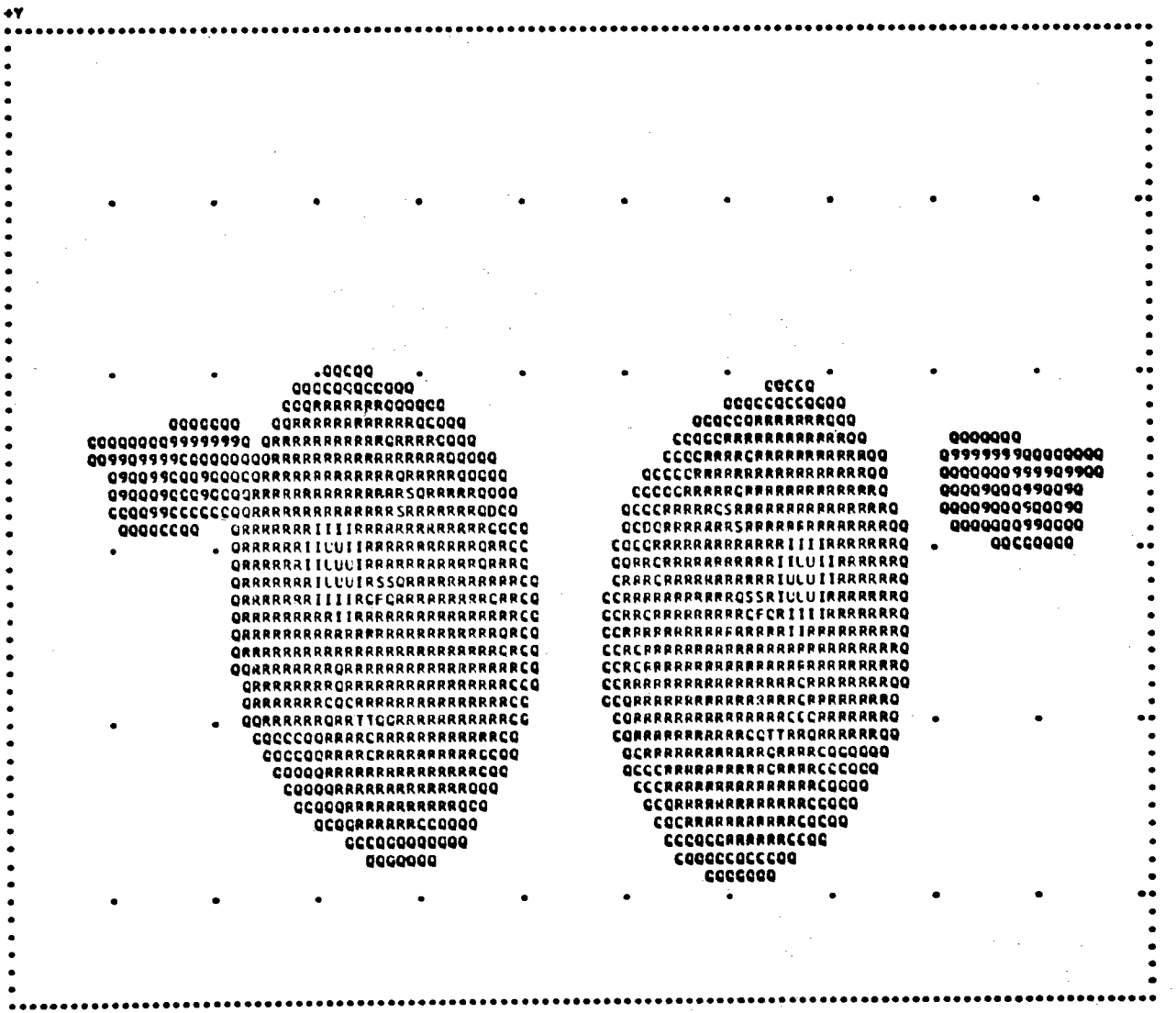
CROSS SECTION 30


+Y
+X
CROSS SECTION 31

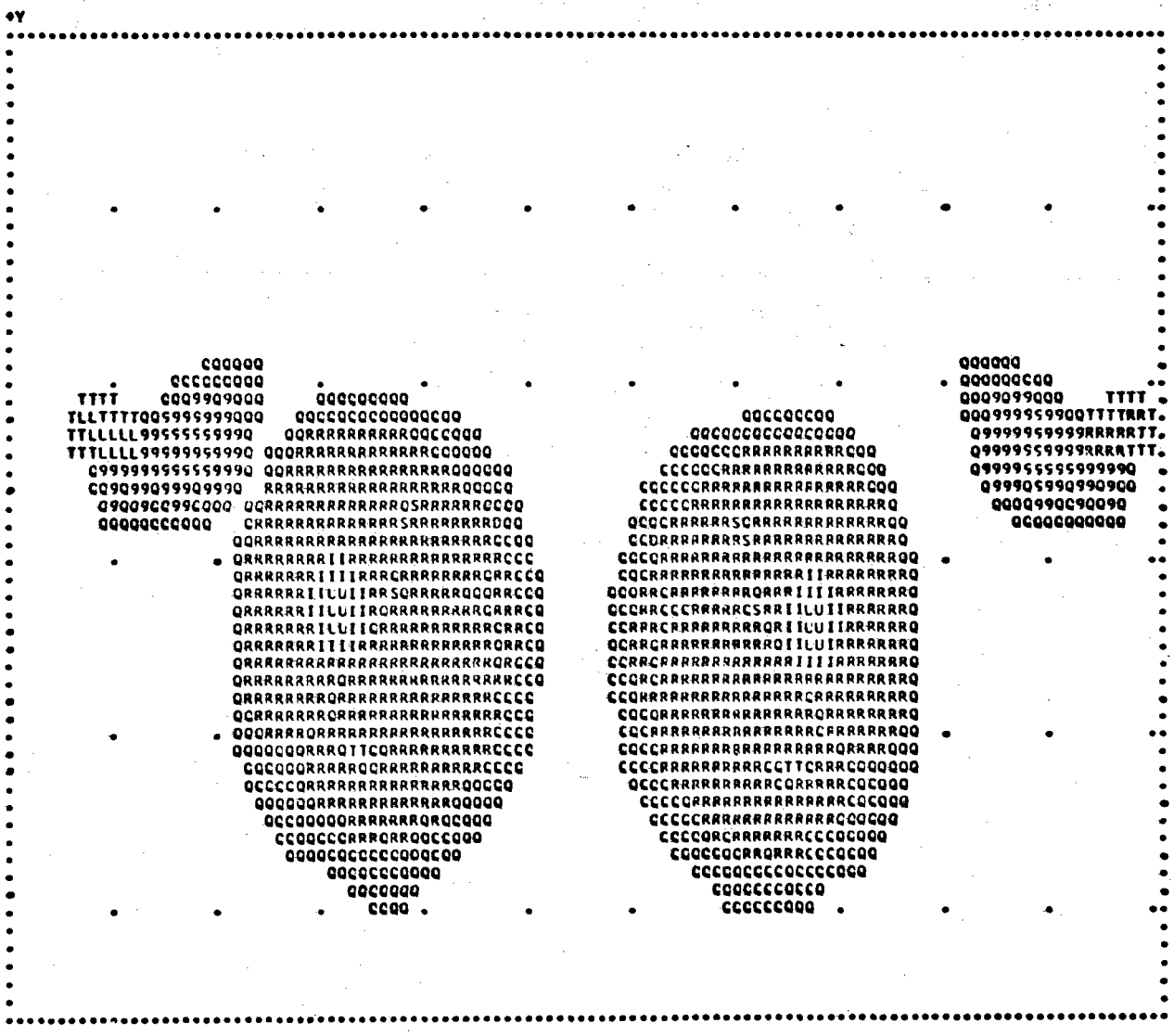

+Y
+X
CROSS SECTION 32


+Y
+X
CROSS SECTION 33

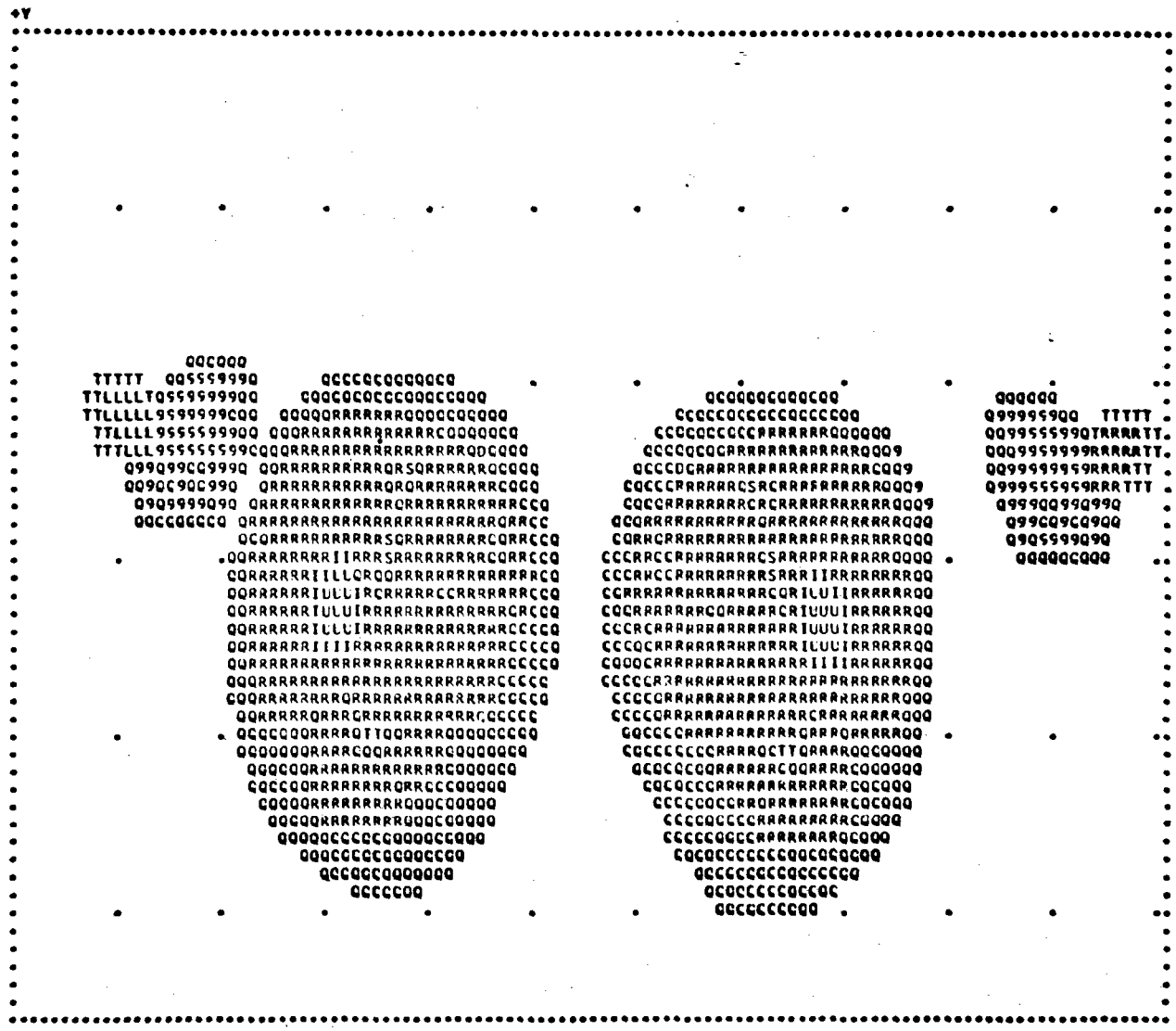

+Y
+X
CROSS SECTION 34



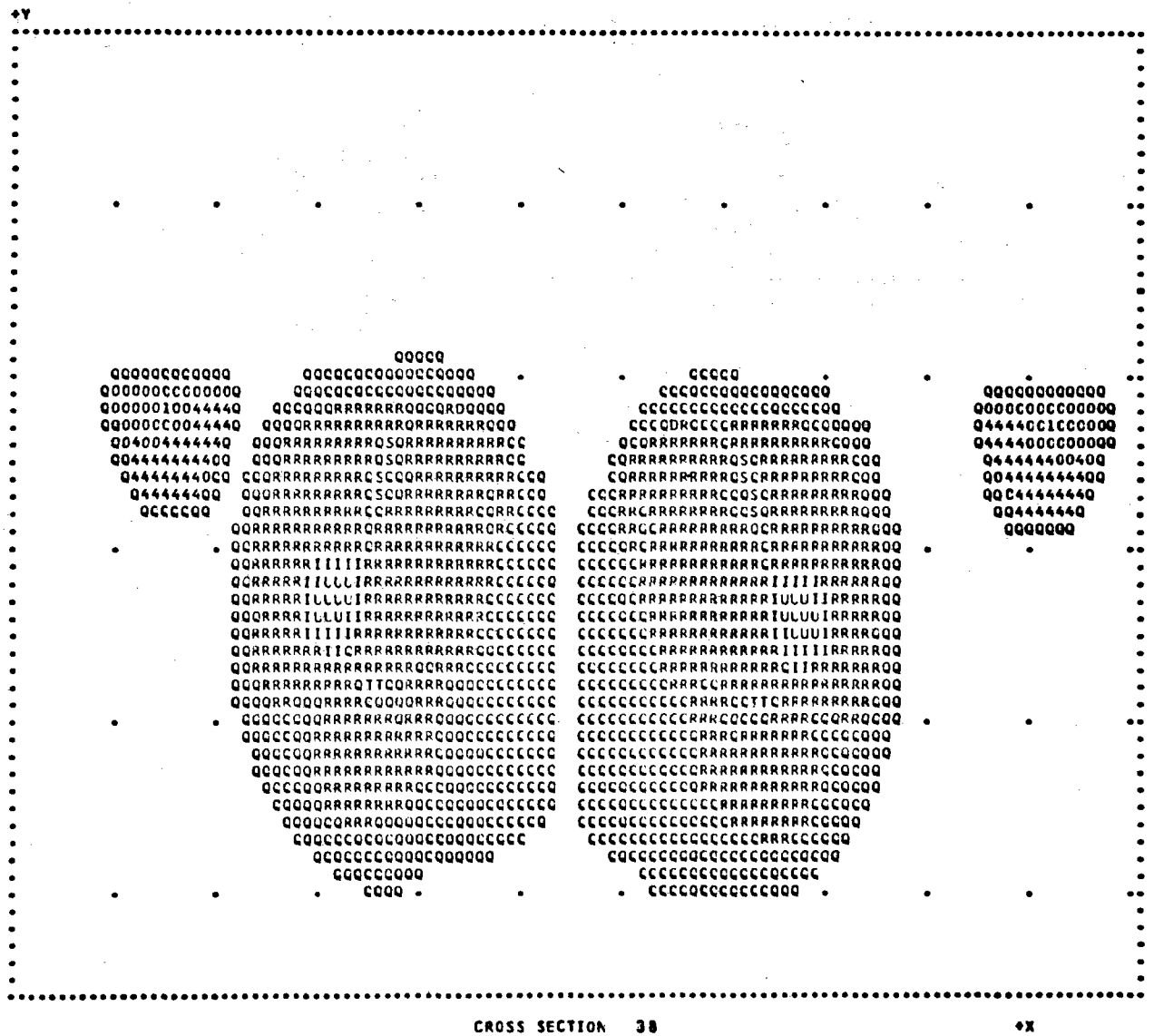
CROSS SECTION 35



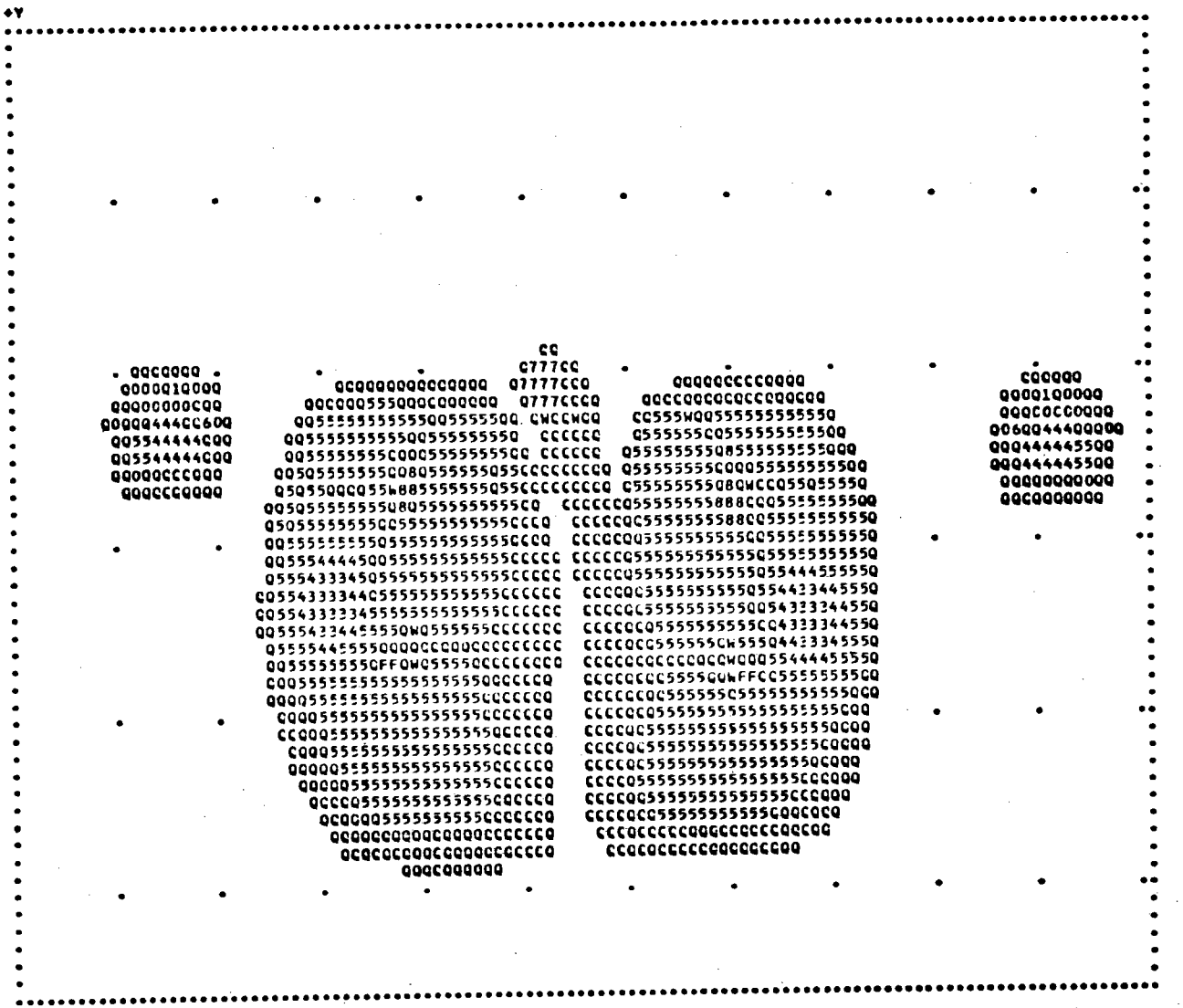
CROSS SECTION 36



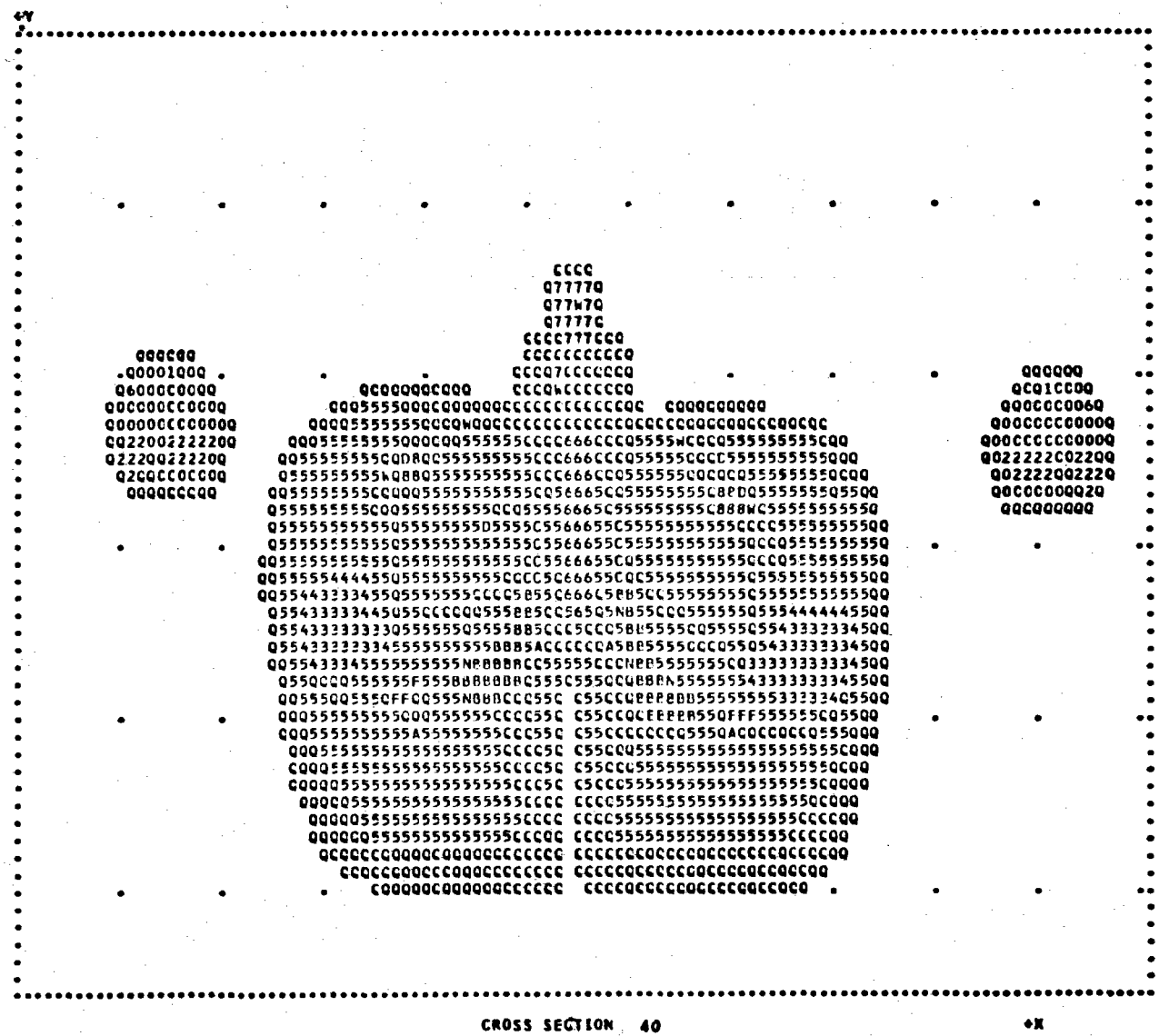
CROSS SECTION 37



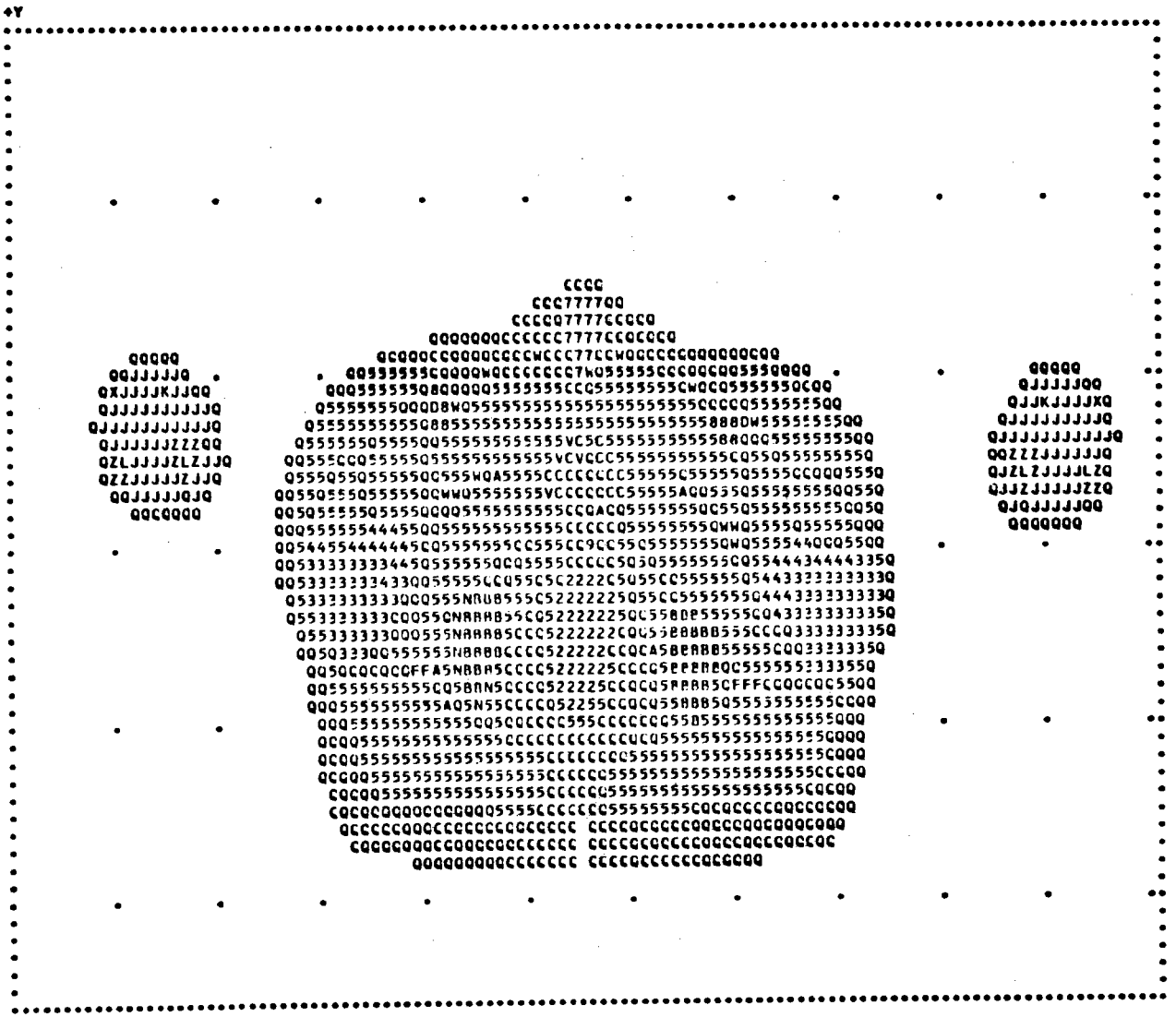
CROSS SECTION 38

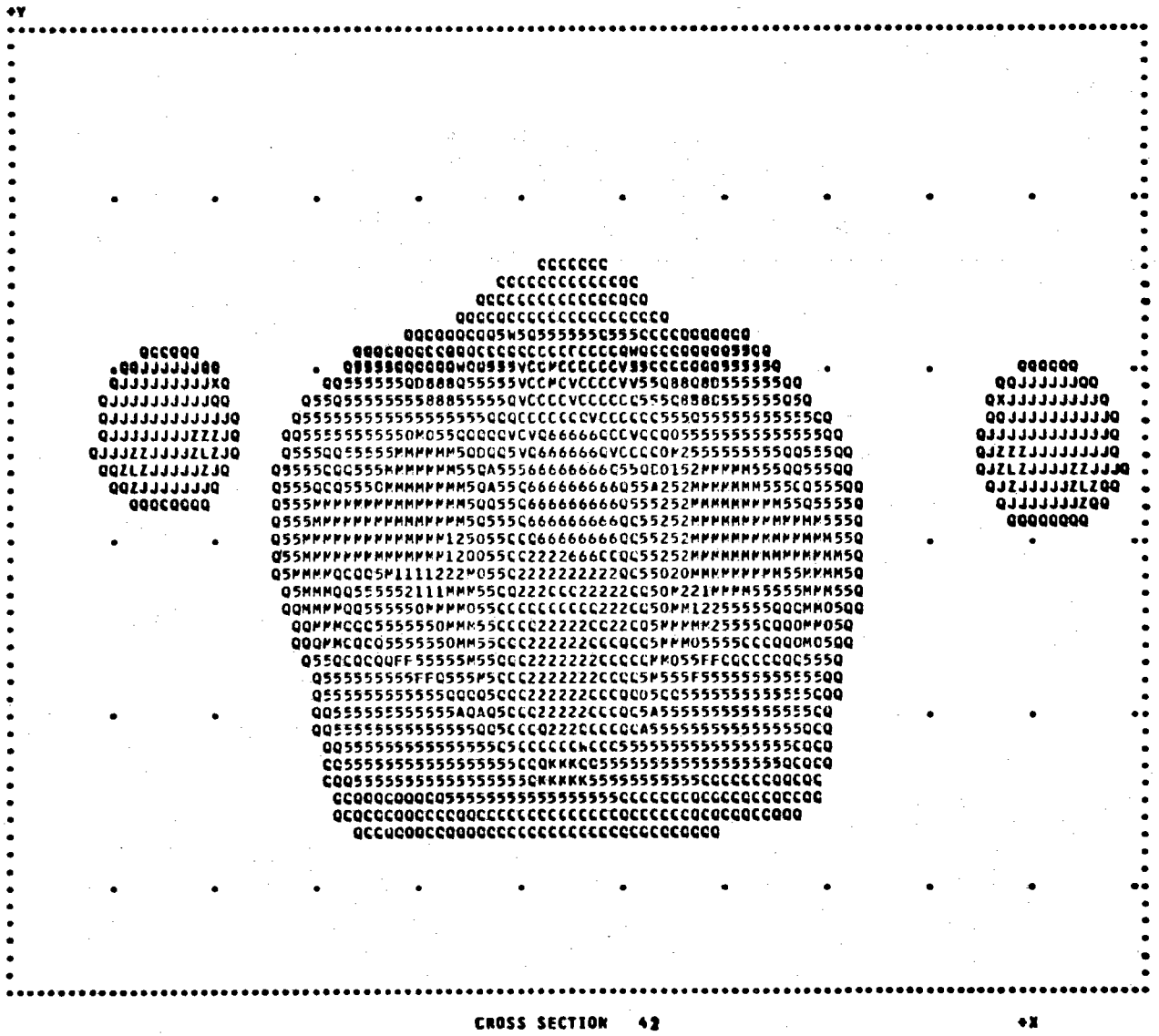


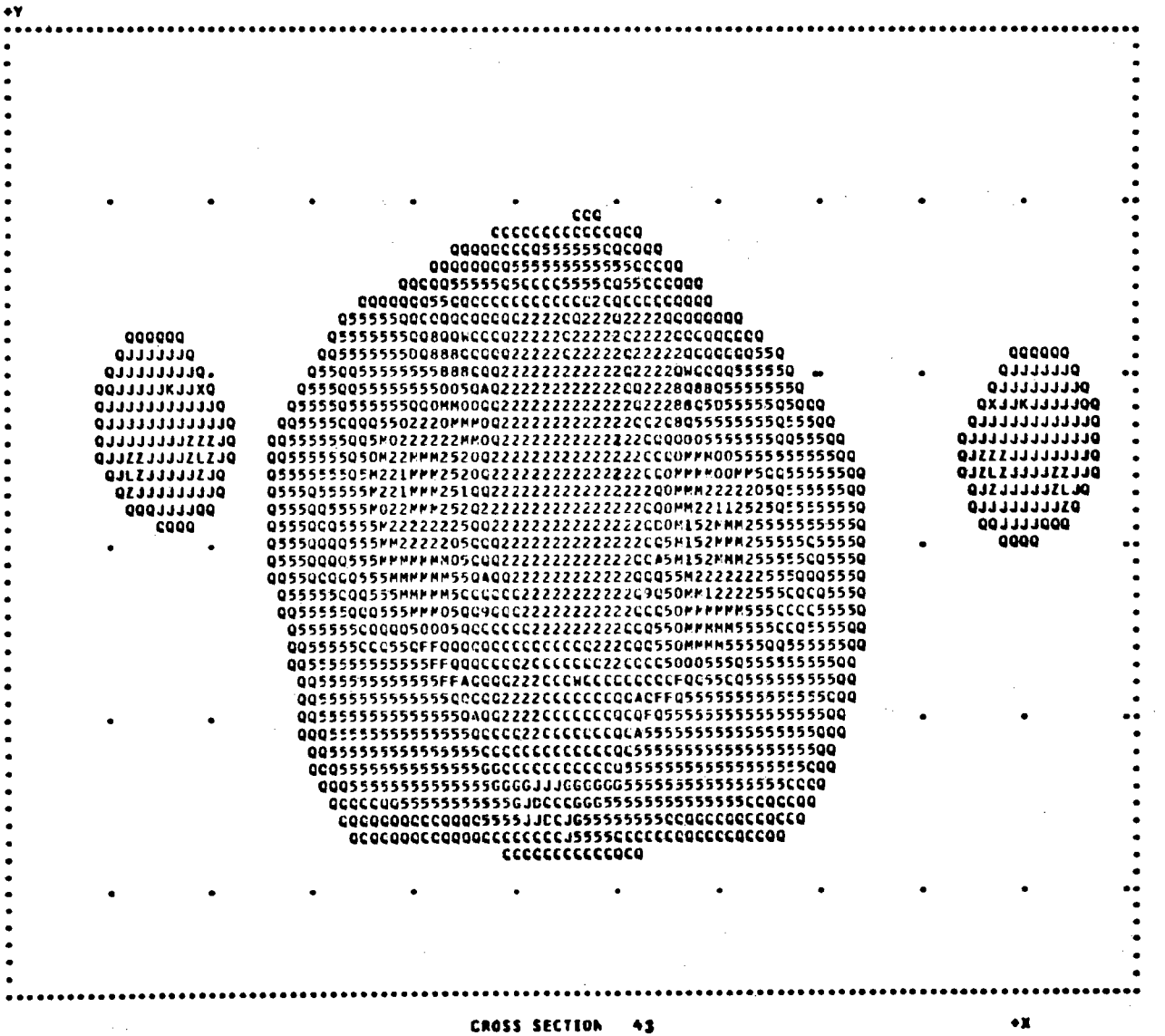
CROSS SECTION 39



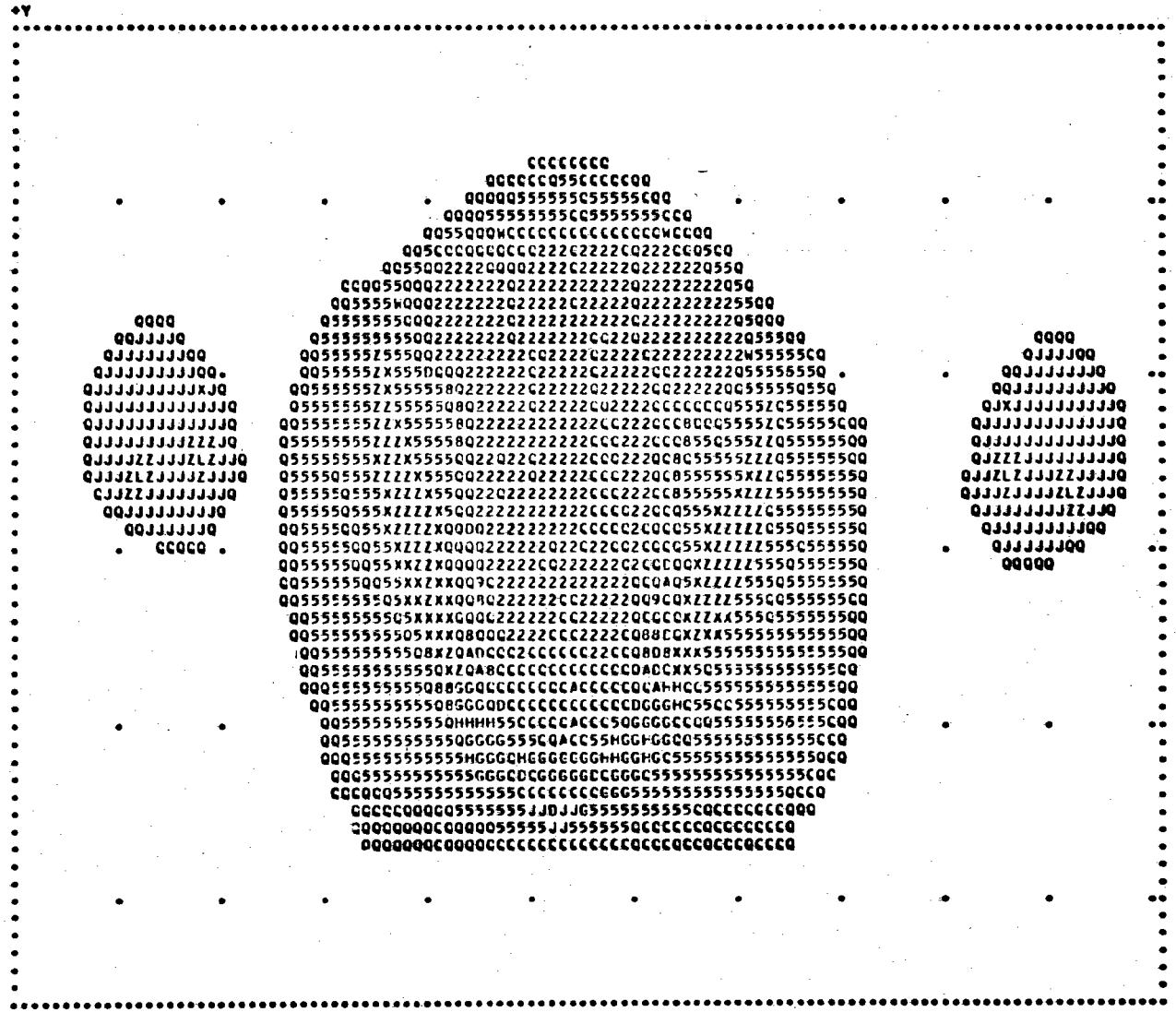
CROSS SECTION 40


+Y
+X
CROSS SECTION 41

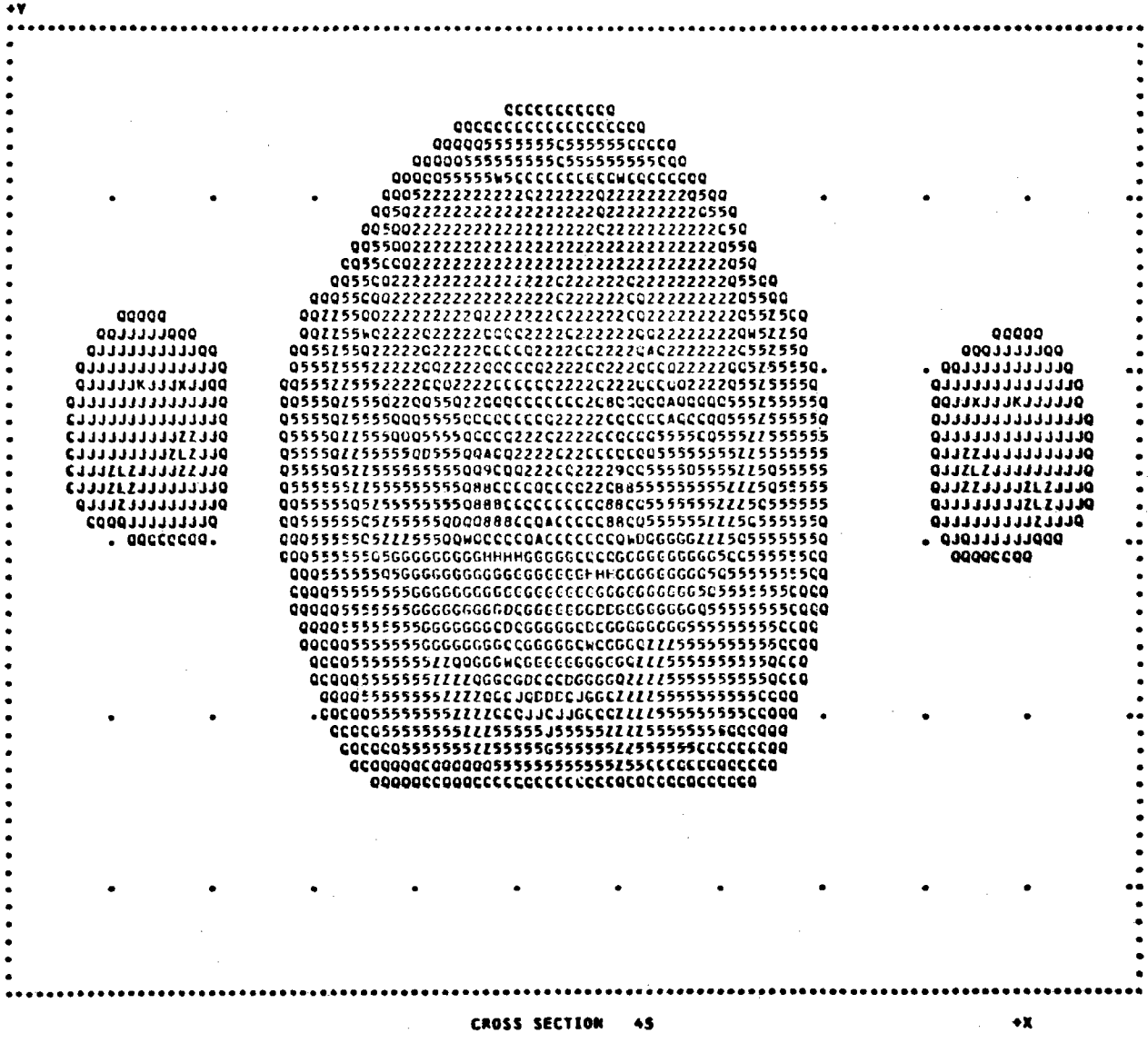

+Y
CROSS SECTION 42
+X

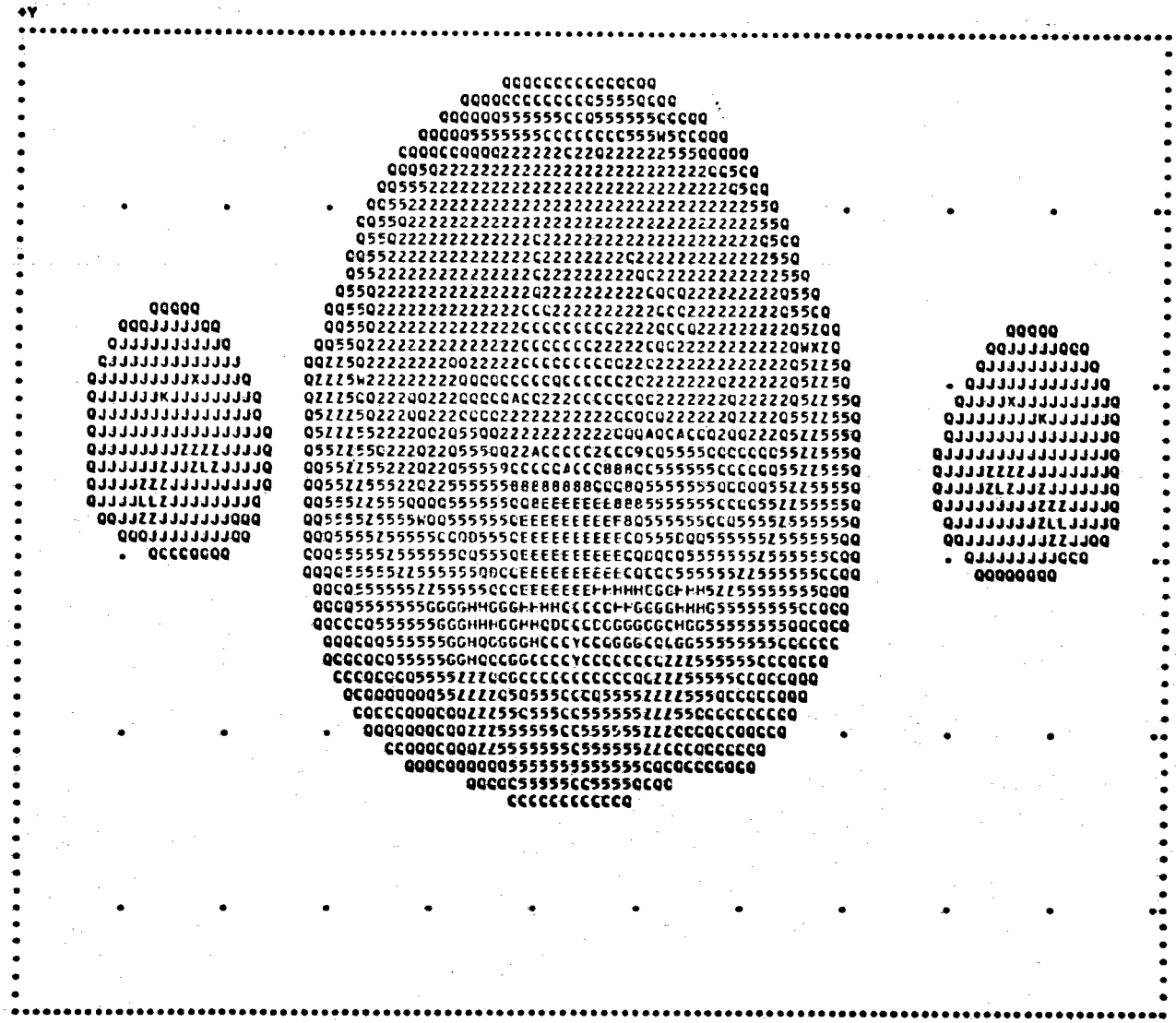


CROSS SECTION 43

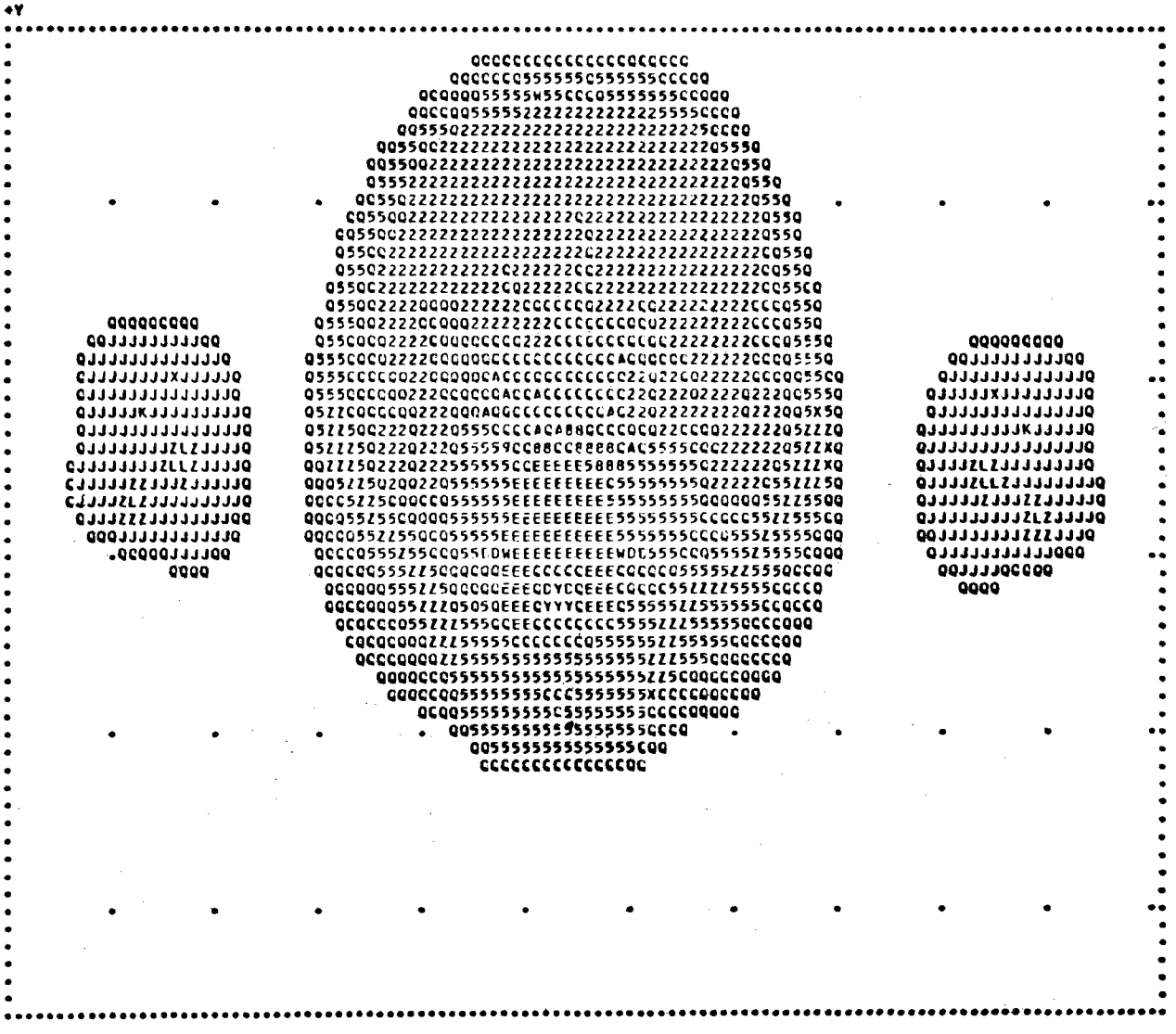


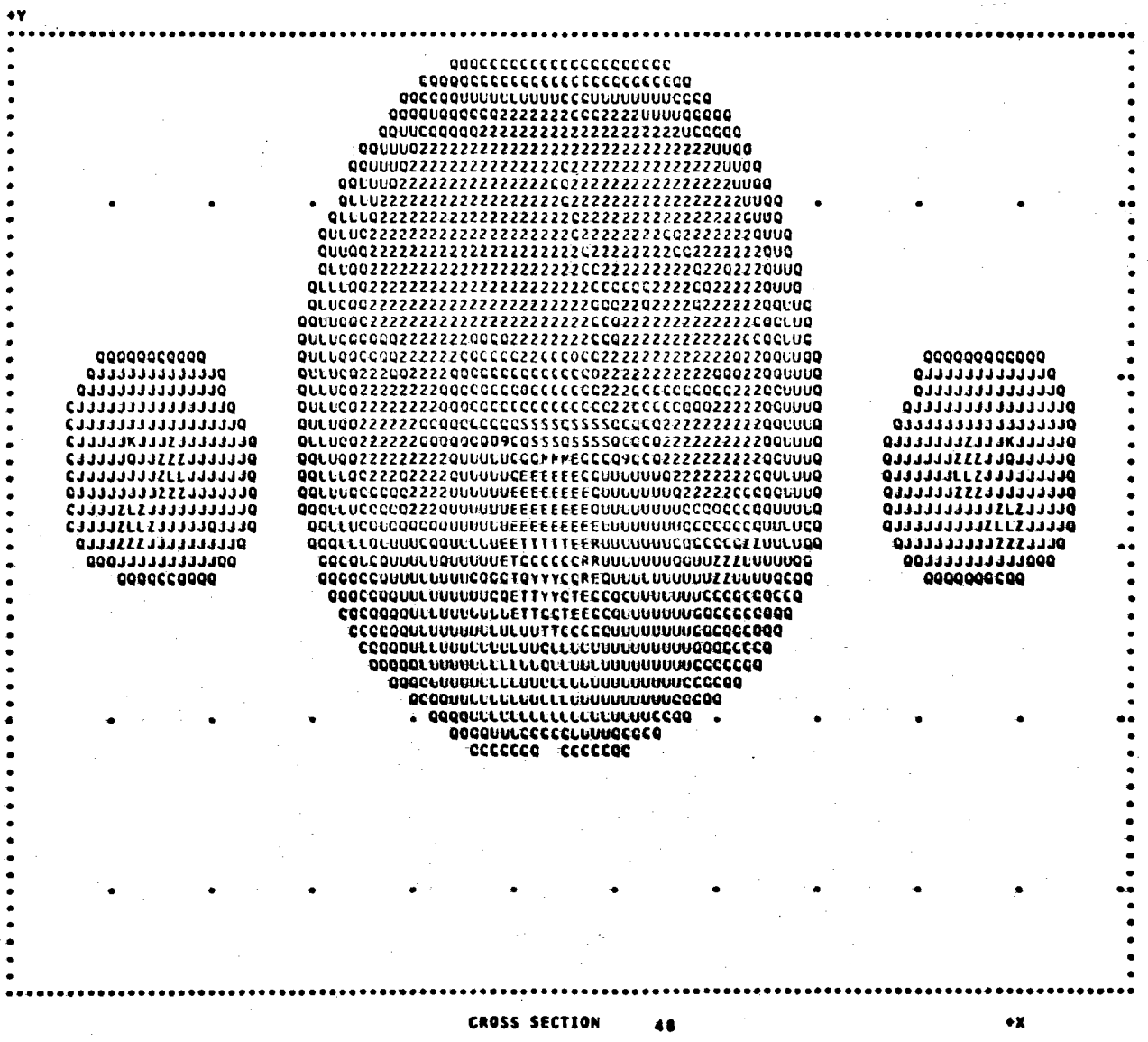
CROSS SECTION 44

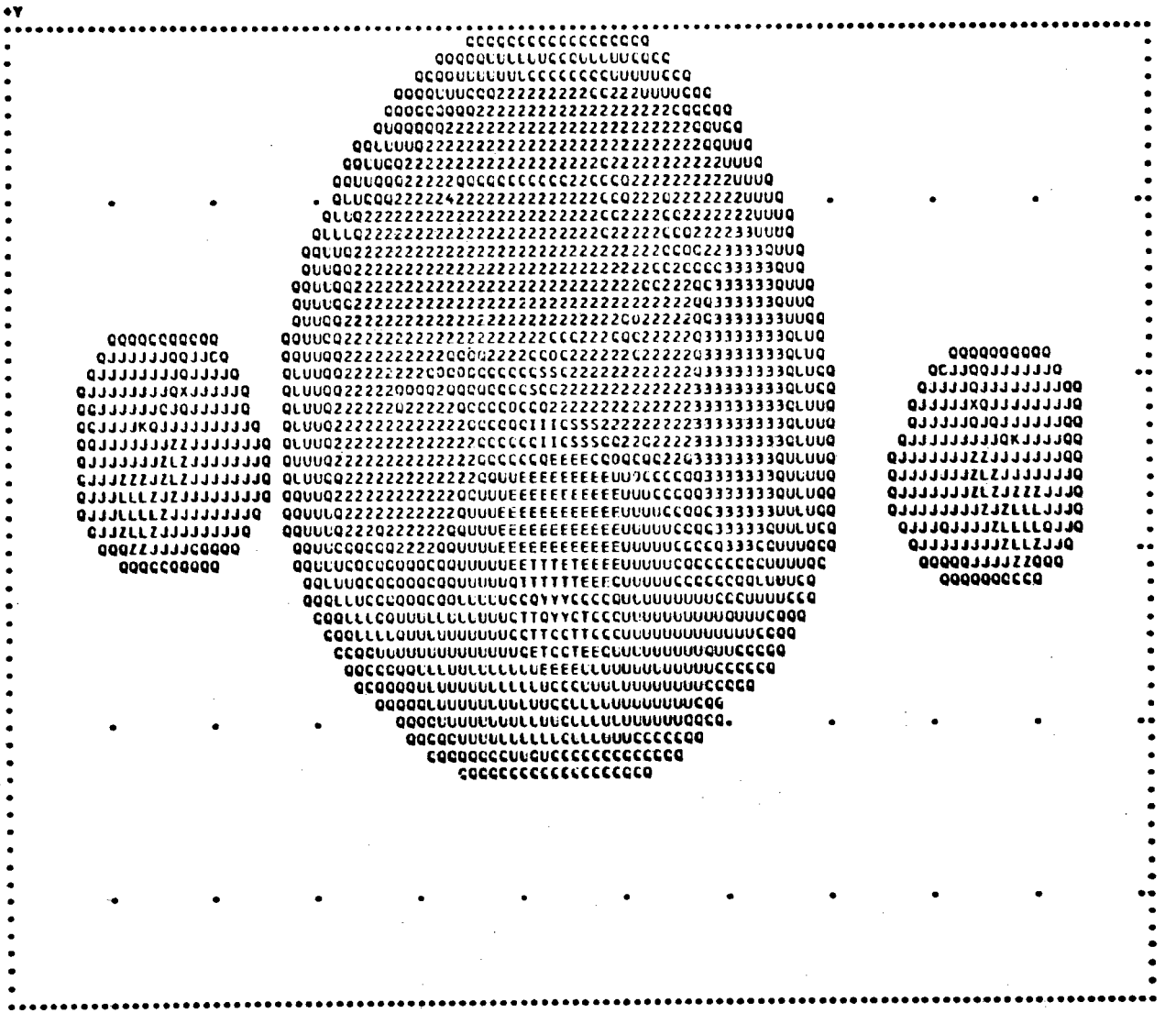

+Y
CROSS SECTION 45
+X

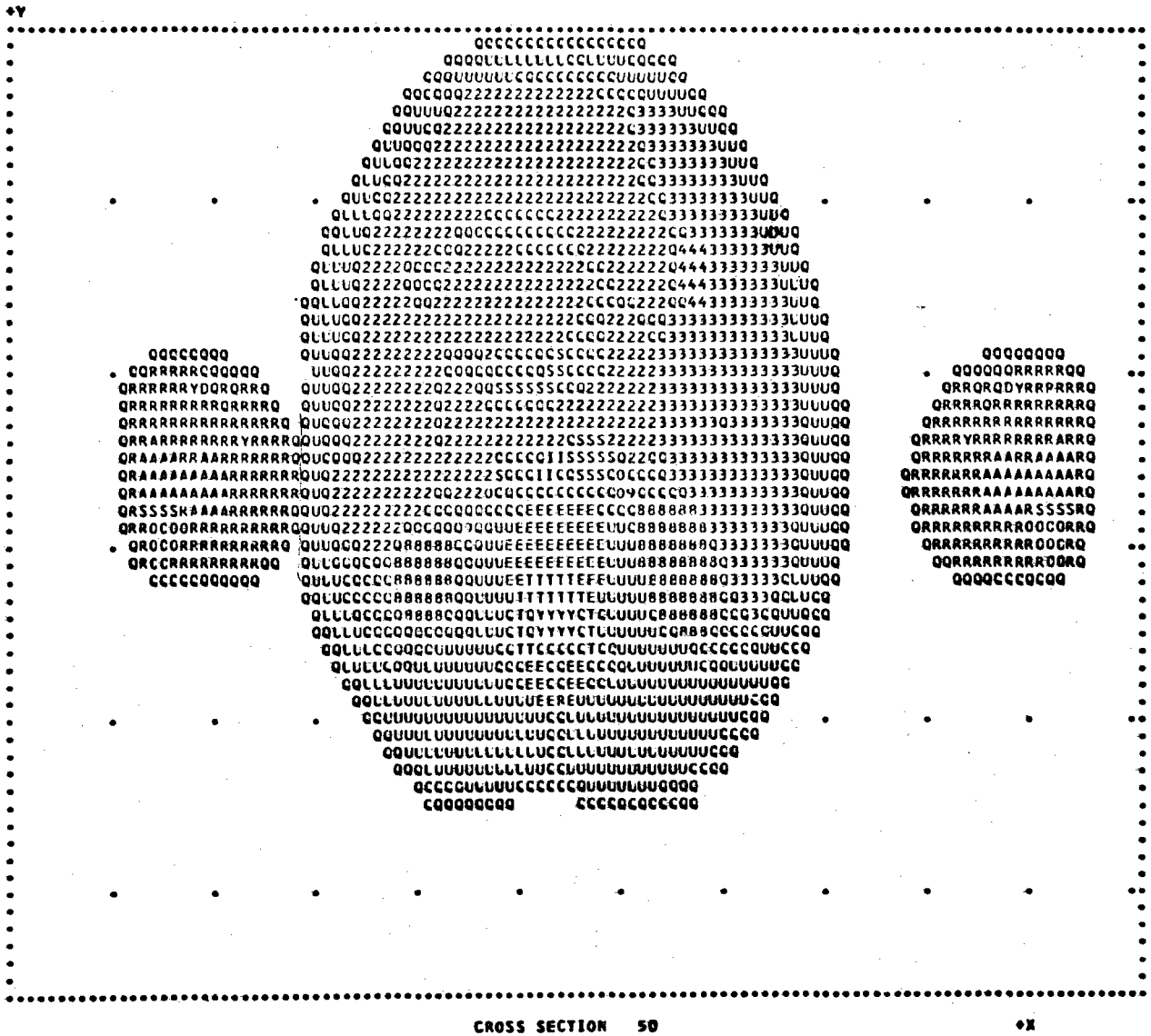


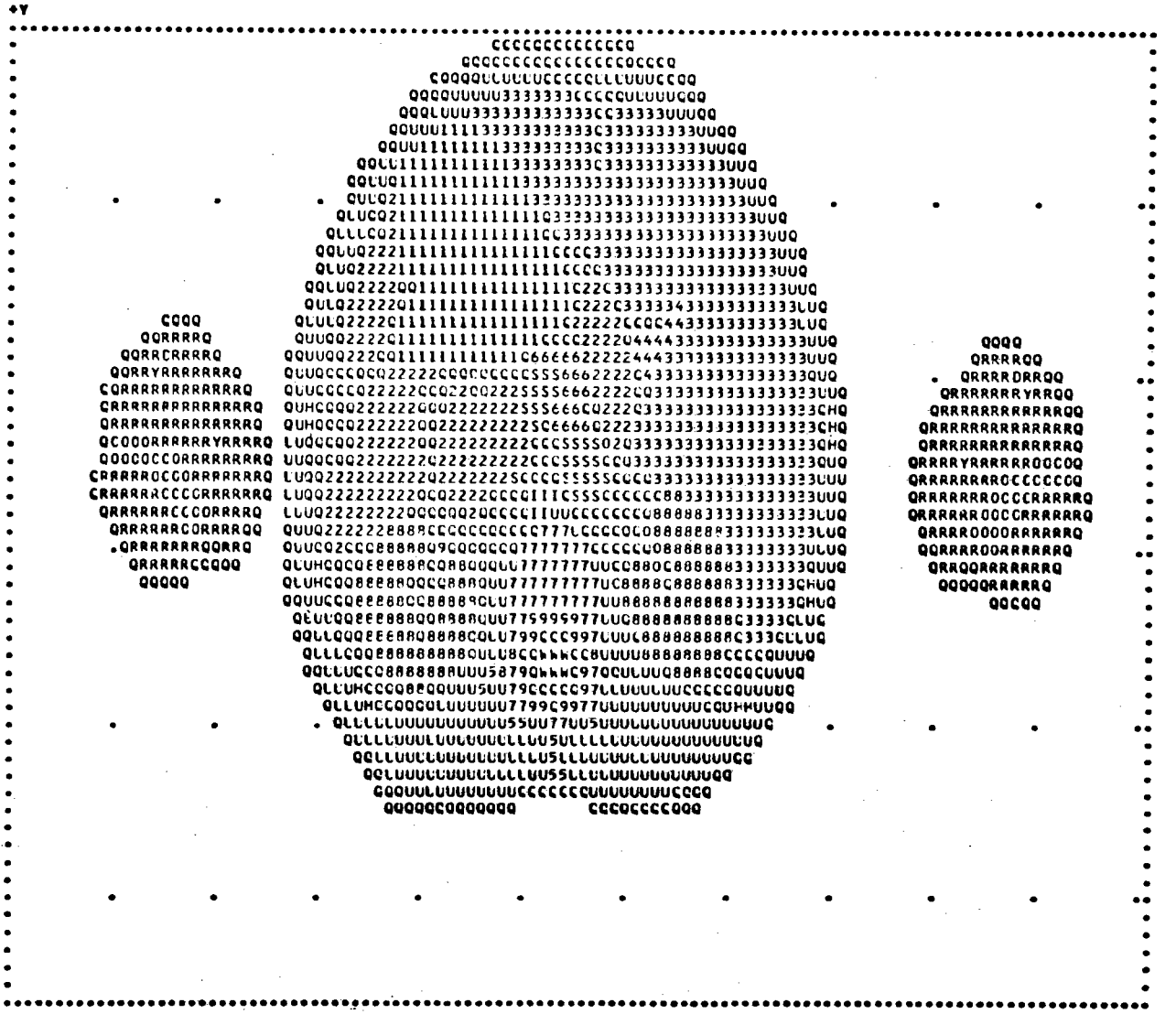
CROSS SECTION 46


+Y
CROSS SECTION 47
+X

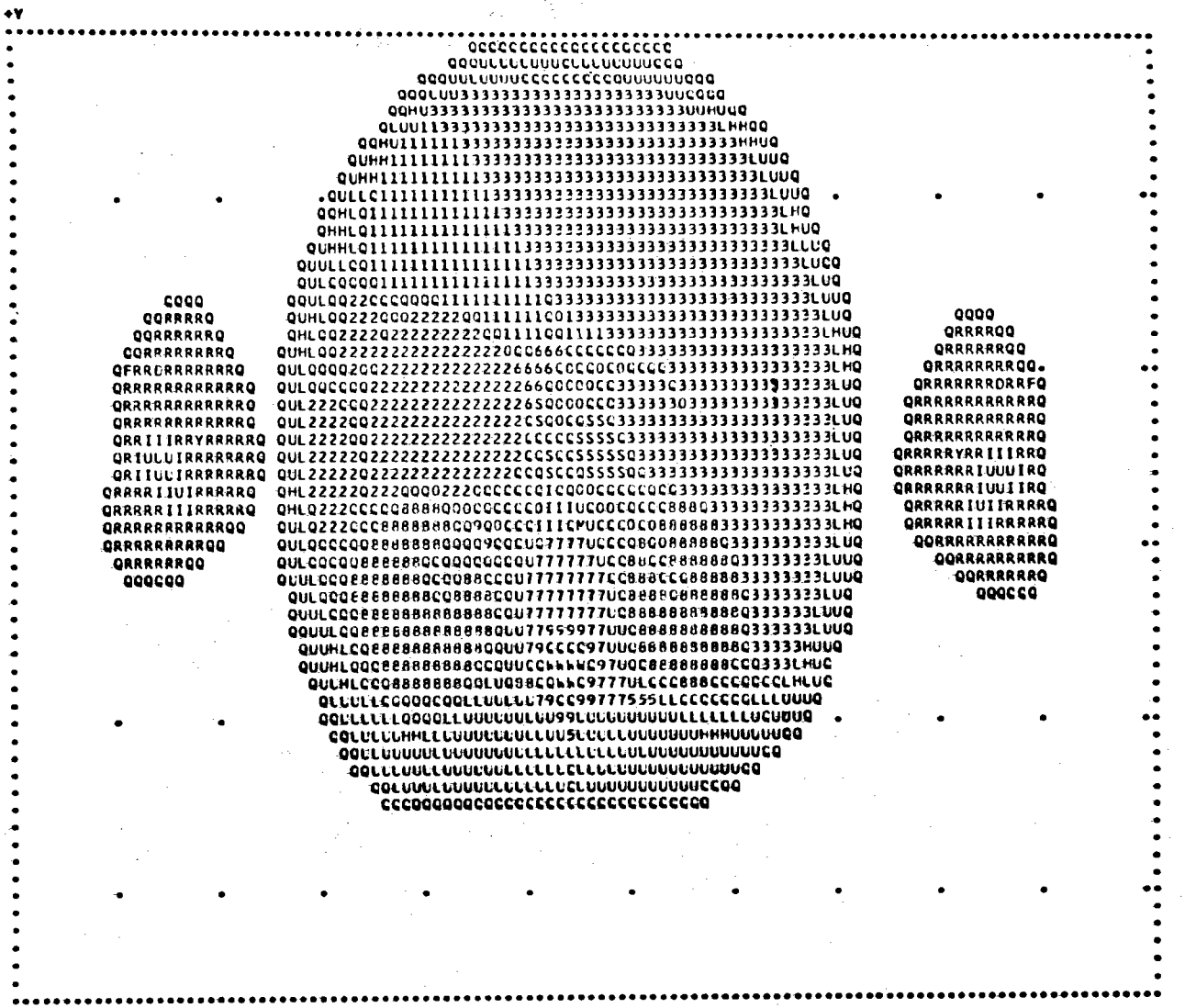

+Y
+X
CROSS SECTION 48


•Y
CROSS SECTION 49
•X

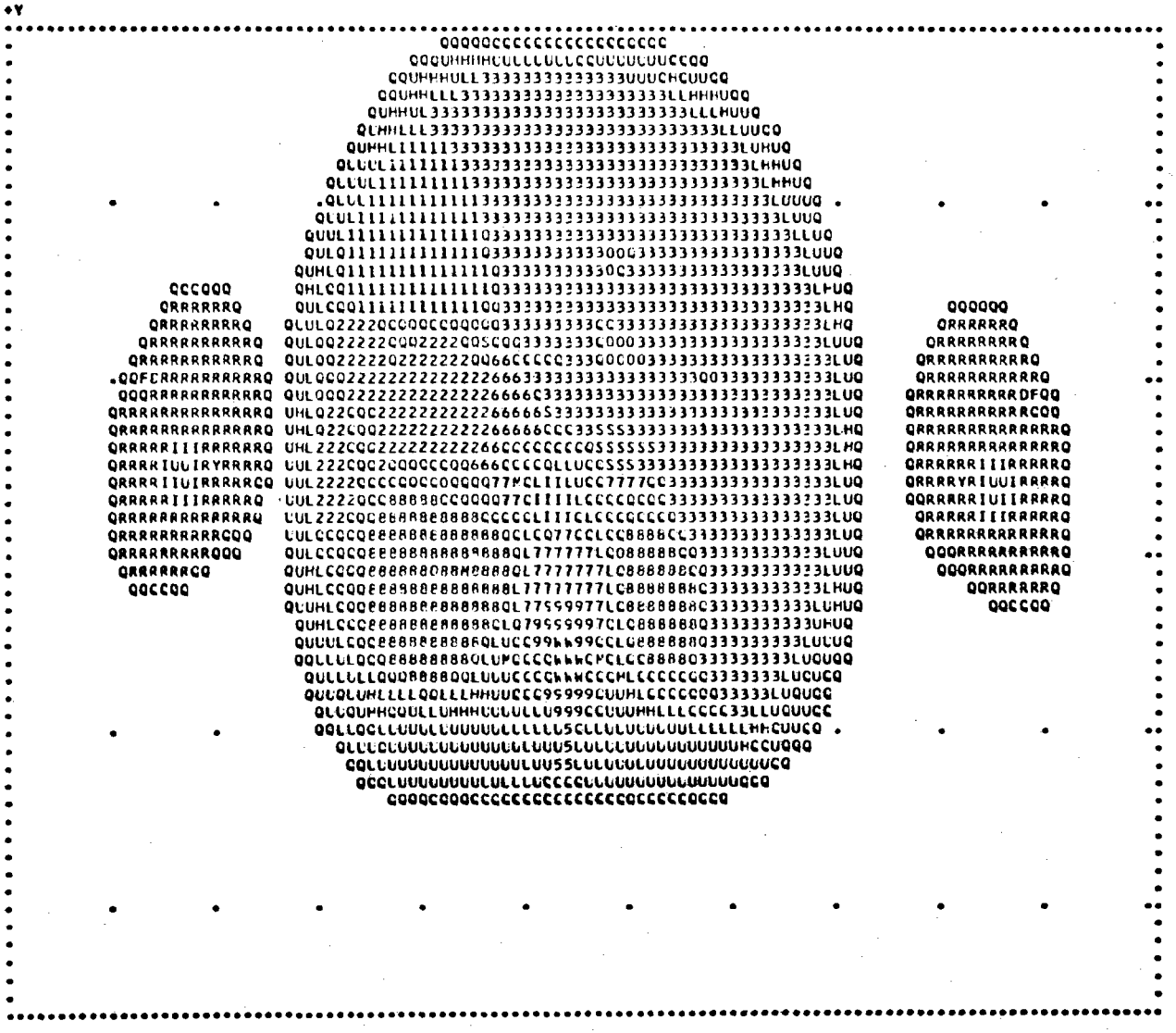

+Y
CROSS SECTION 50
+X

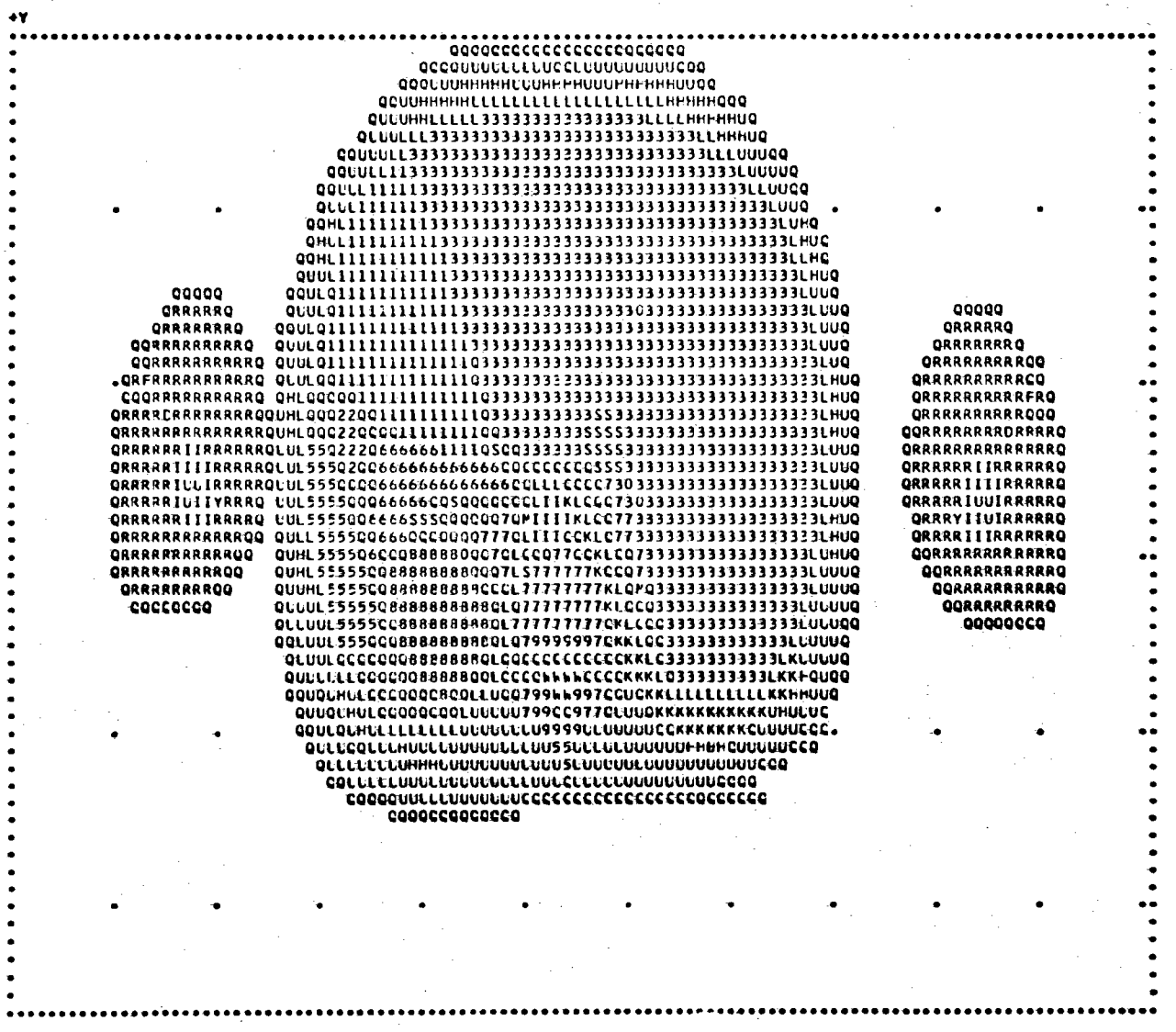


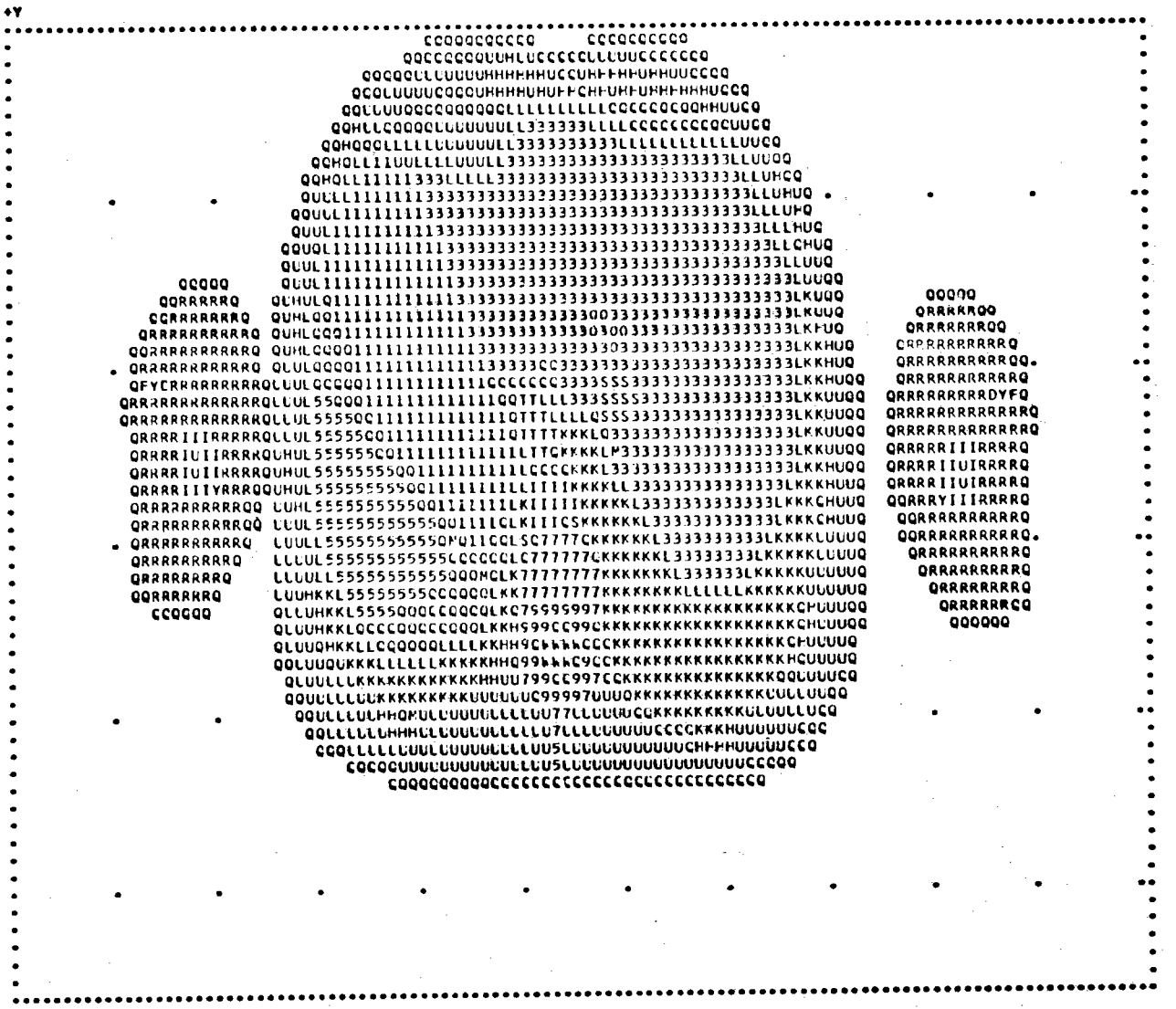
CROSS SECTION 51



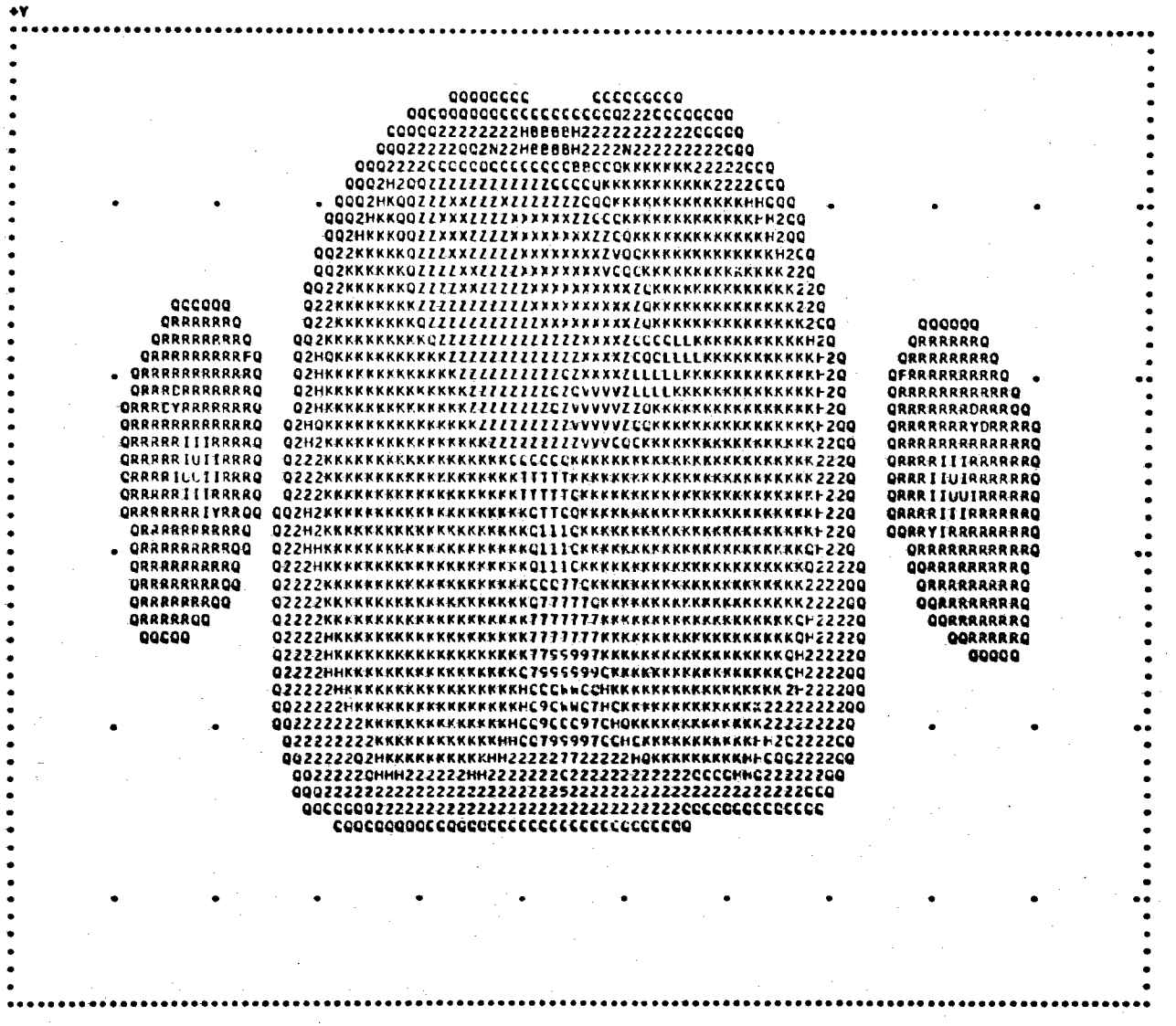
CROSS SECTION 52

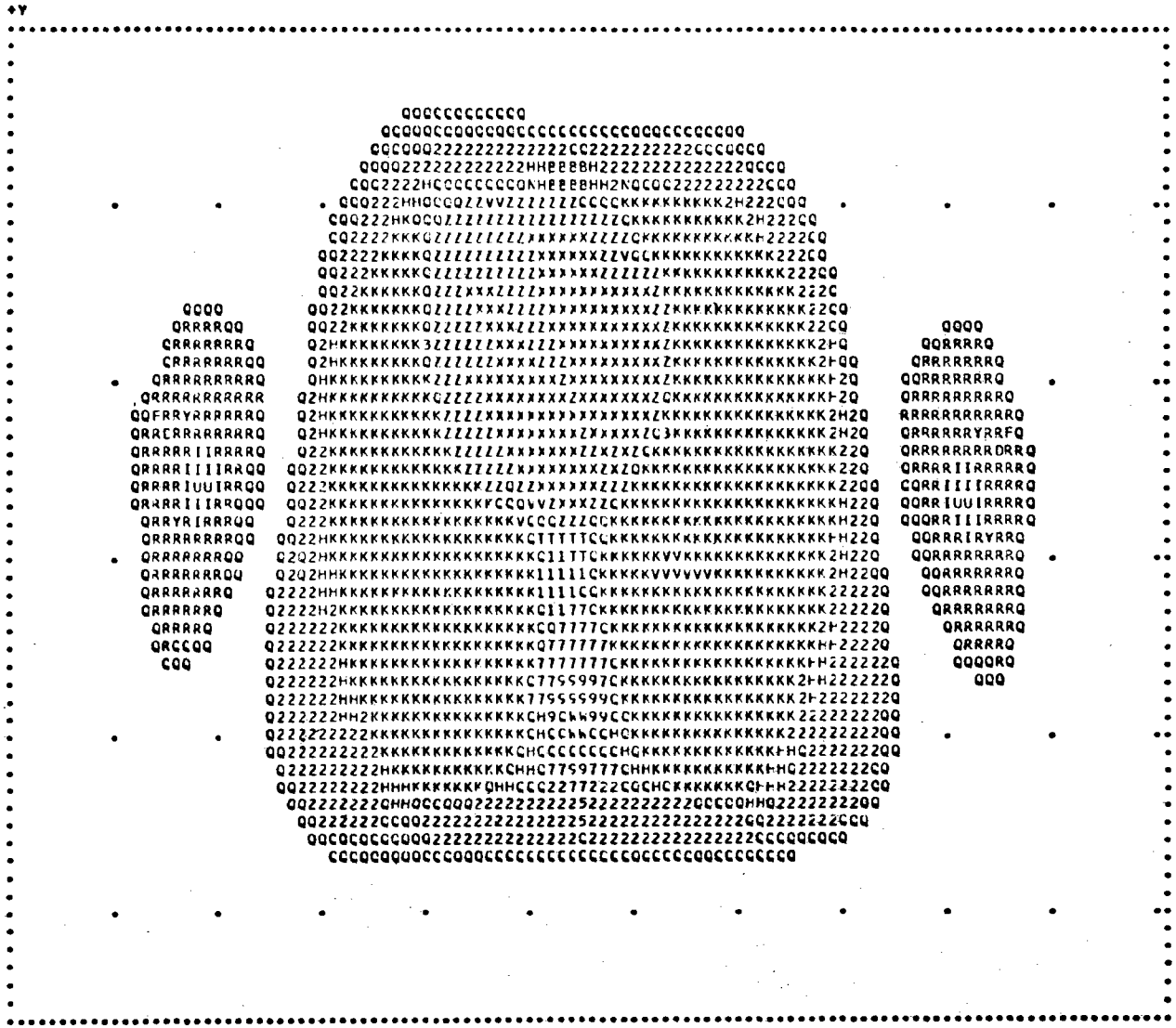

+Y
CROSS SECTION 53
+X


+Y
CROSS SECTION 54
+X



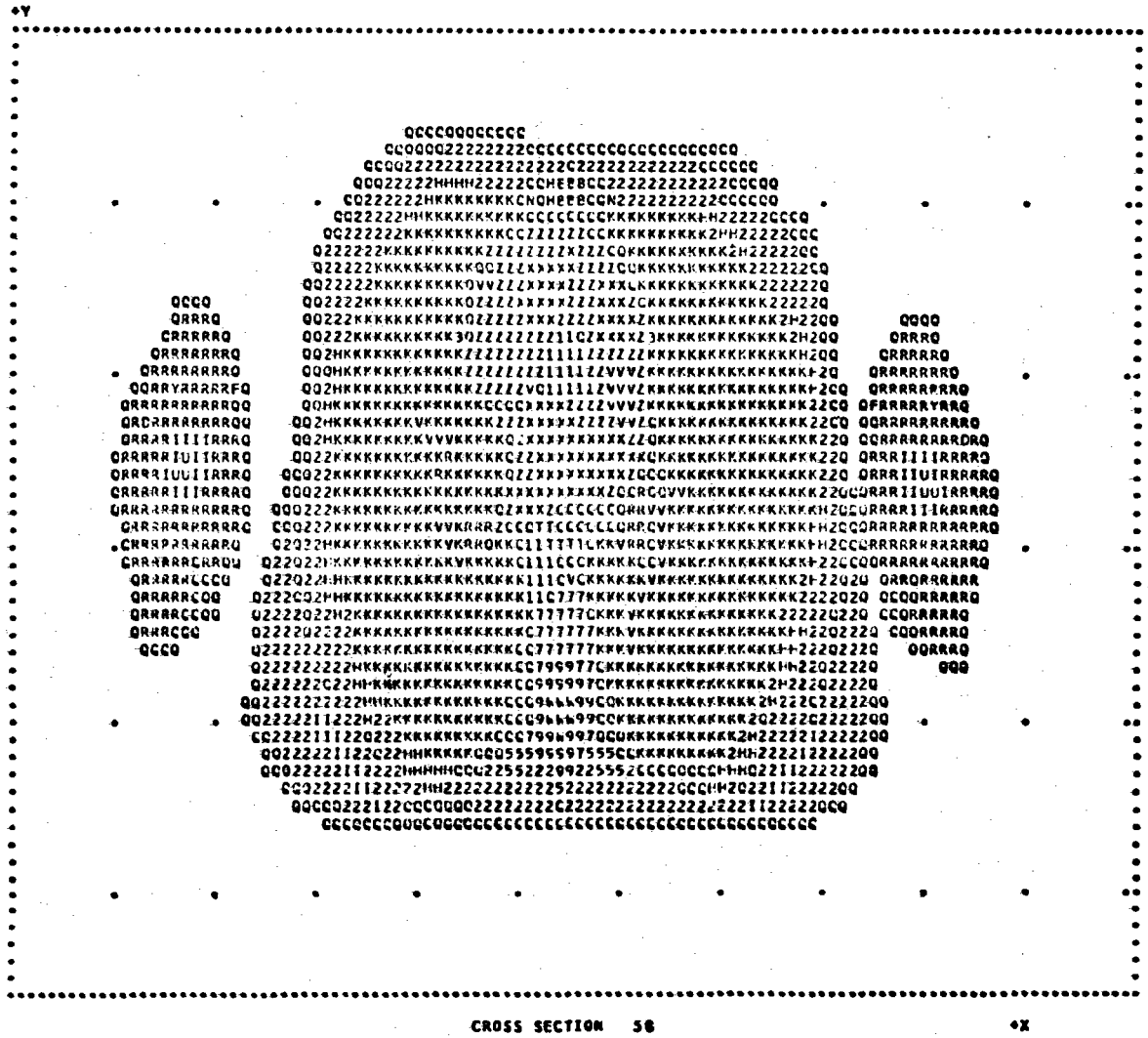
CROSS SECTION 55

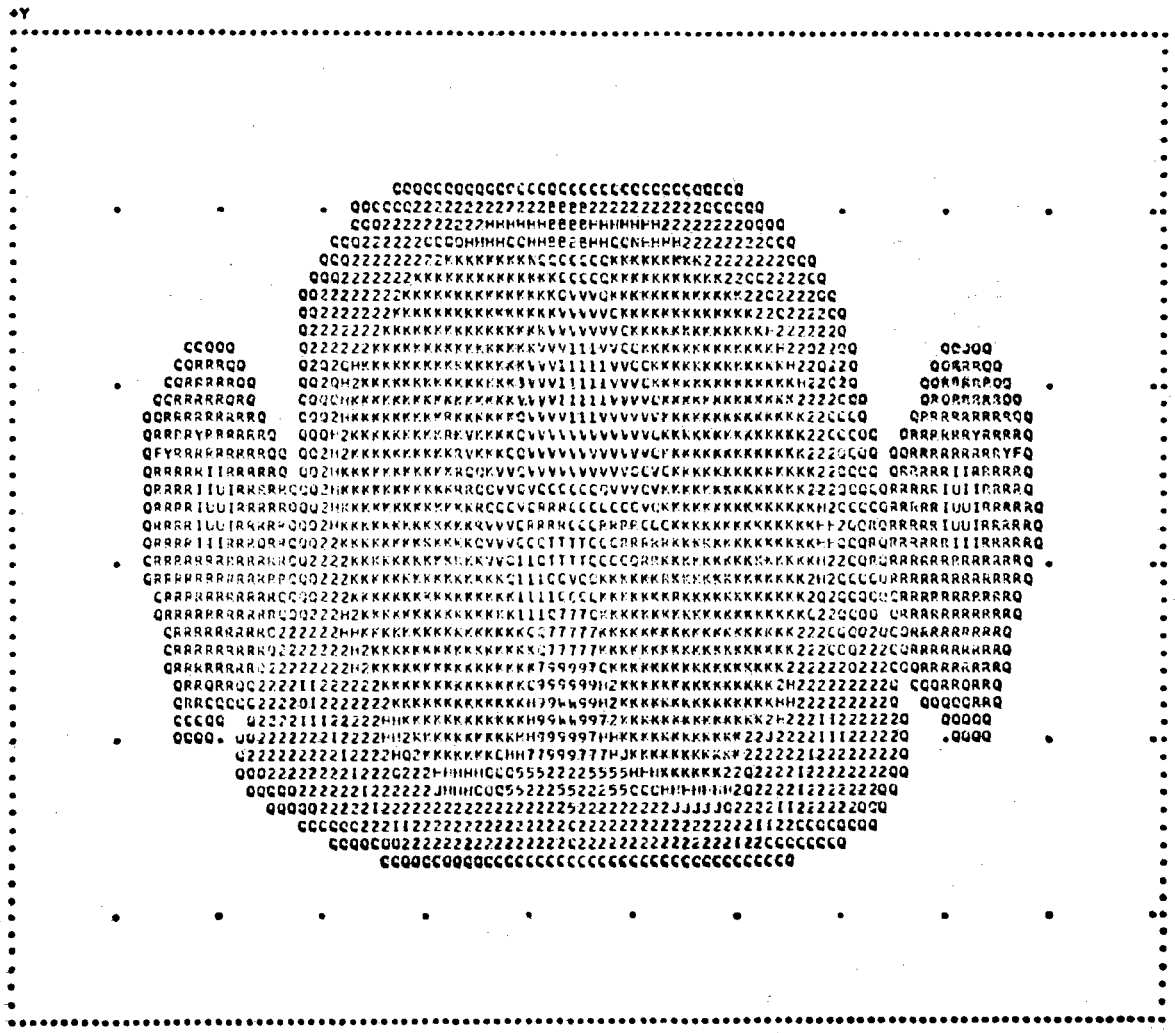

+Y
CROSS SECTION 56
+X



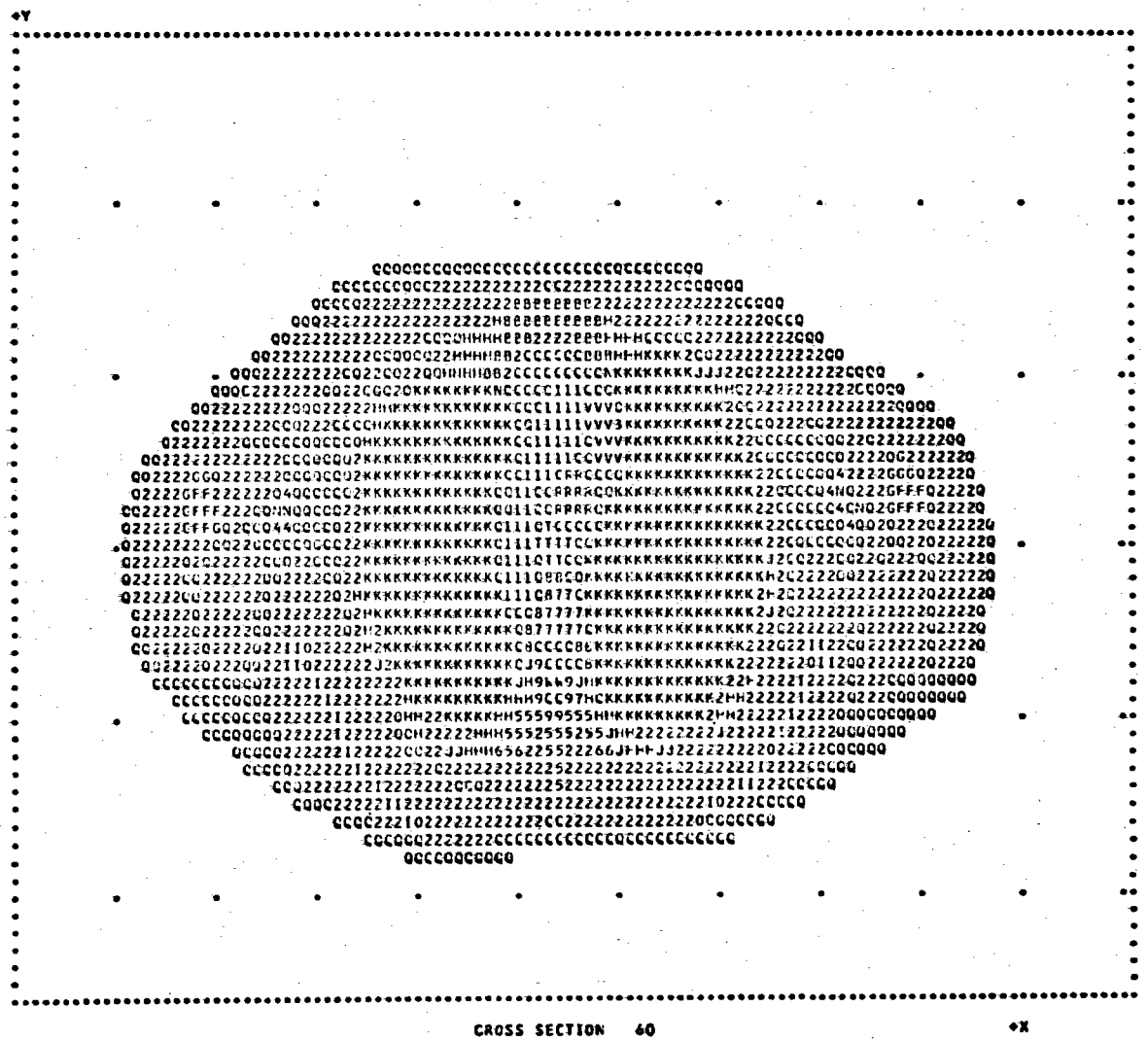
CROSS SECTION 57

PROPERTY OF U.S. ARMY
STINFO BRANCH
BRL, APG, MD. 21005

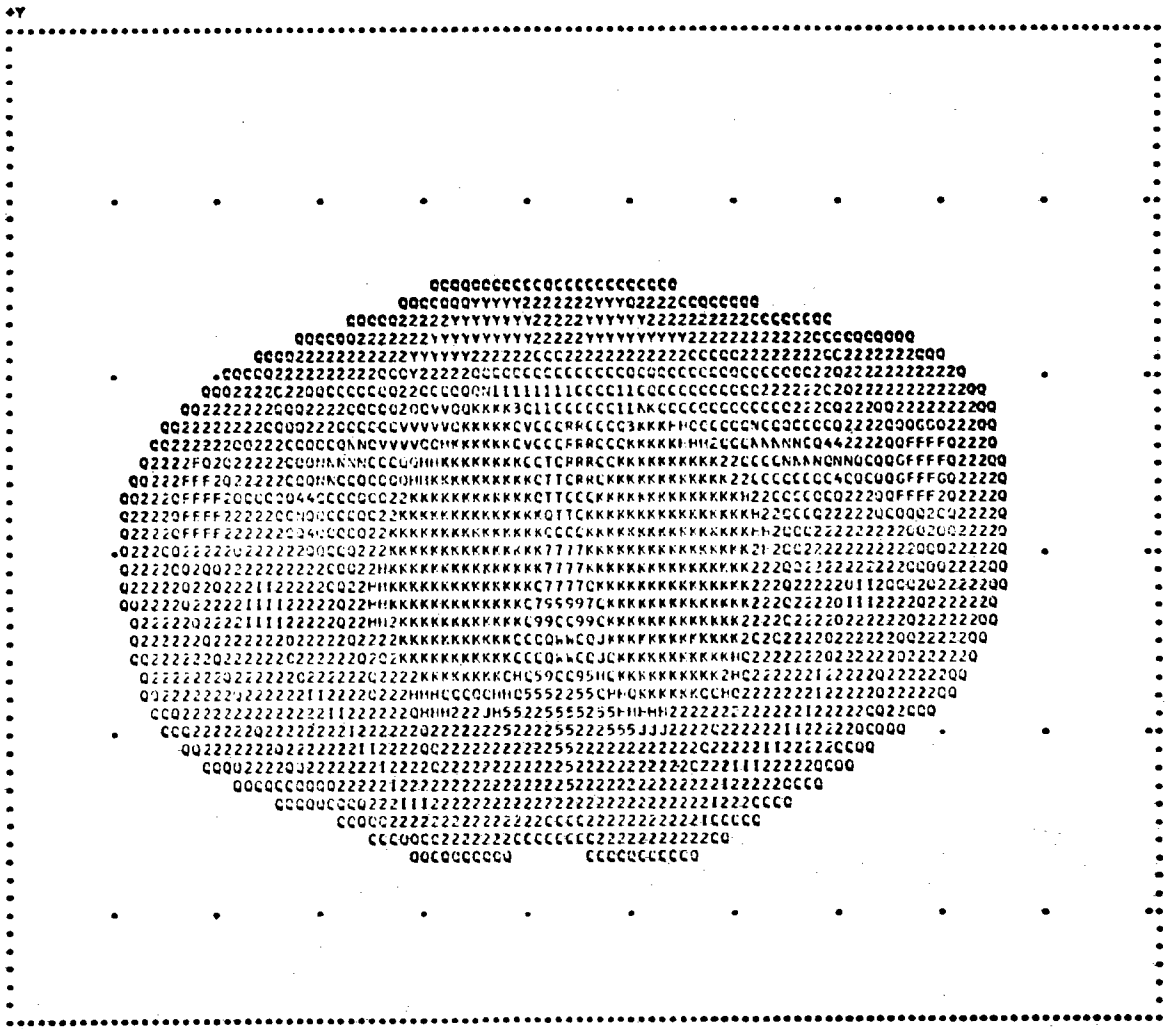

Y
X
CROSS SECTION 58



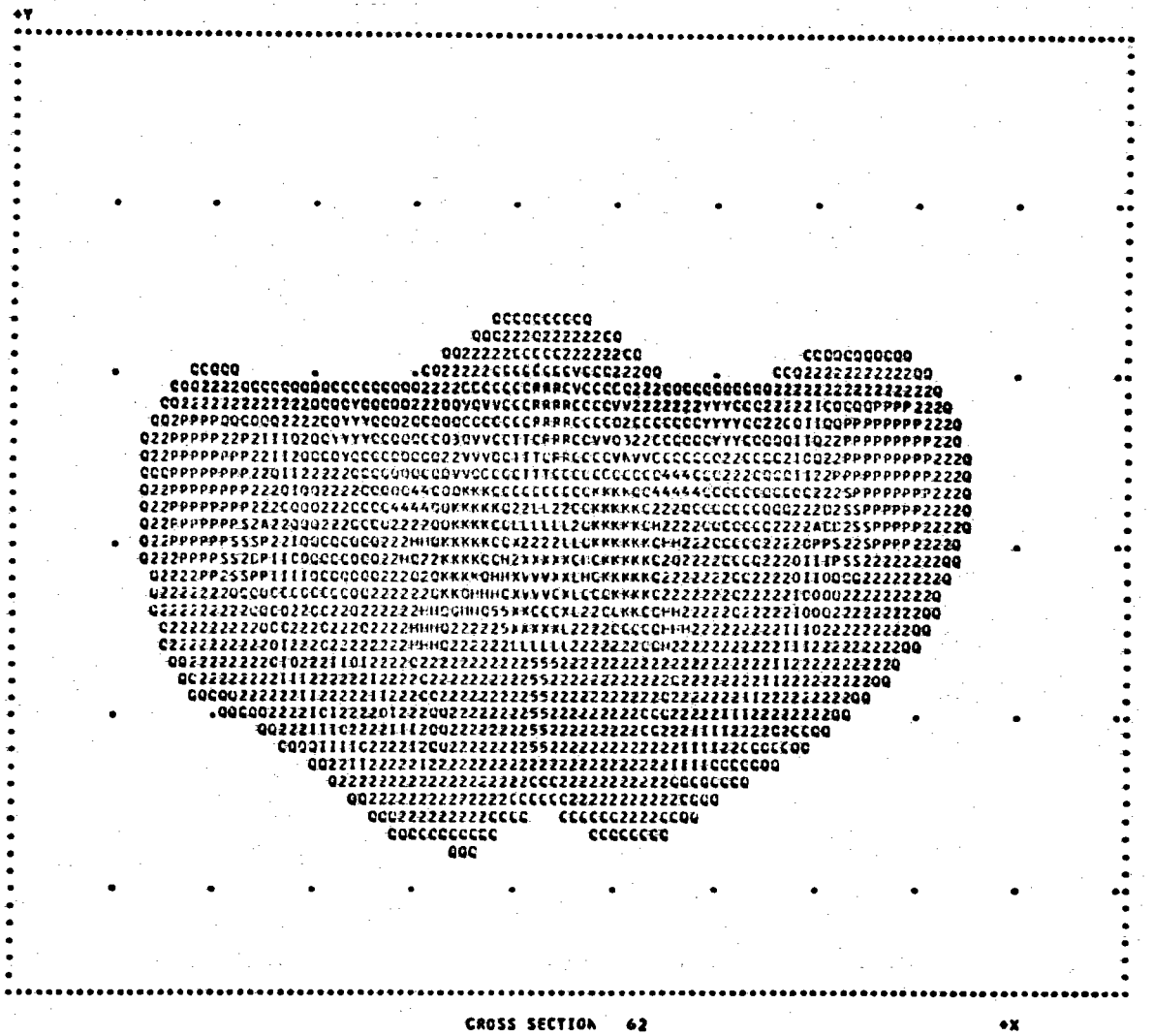
CROSS SECTION 89

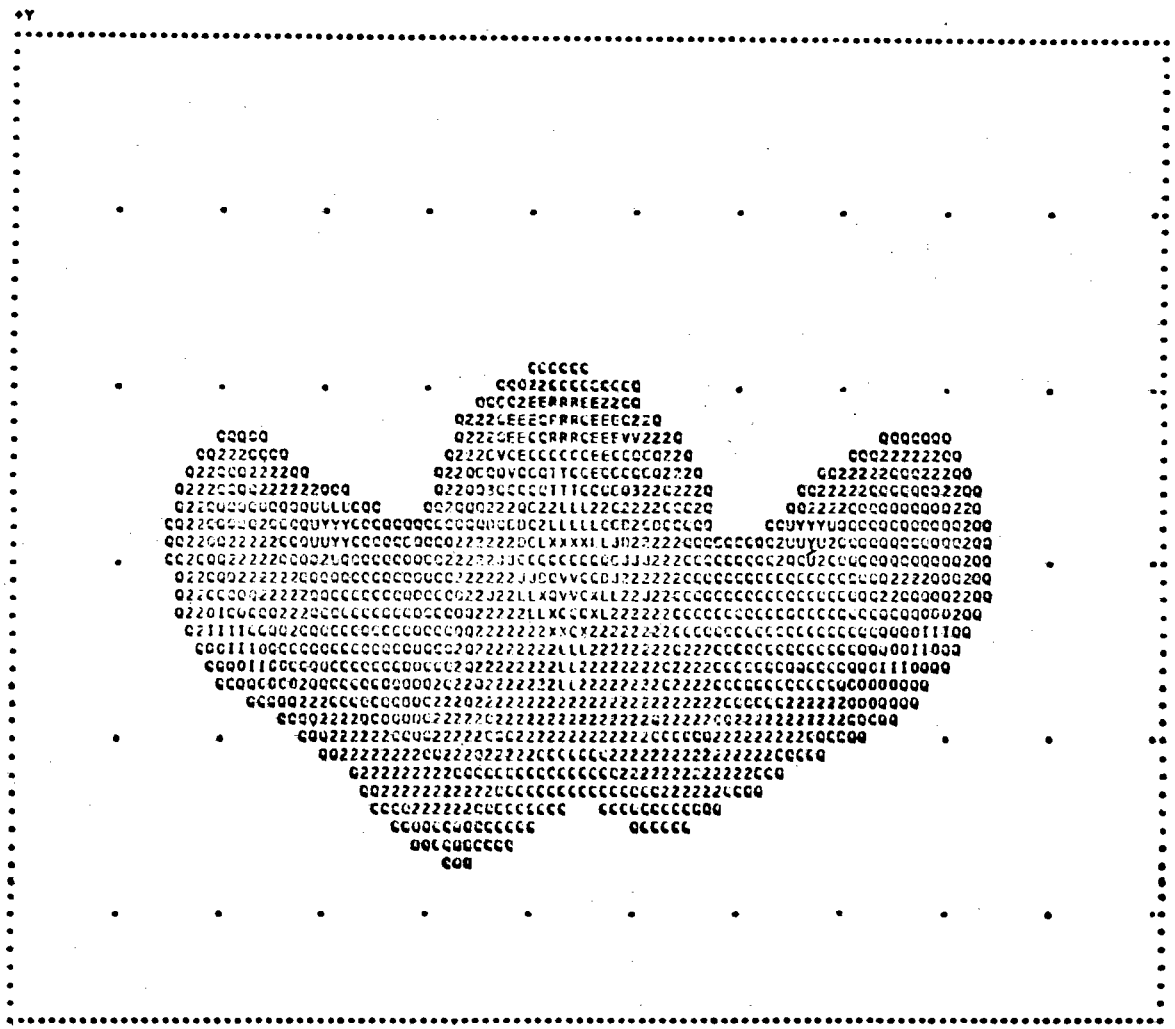


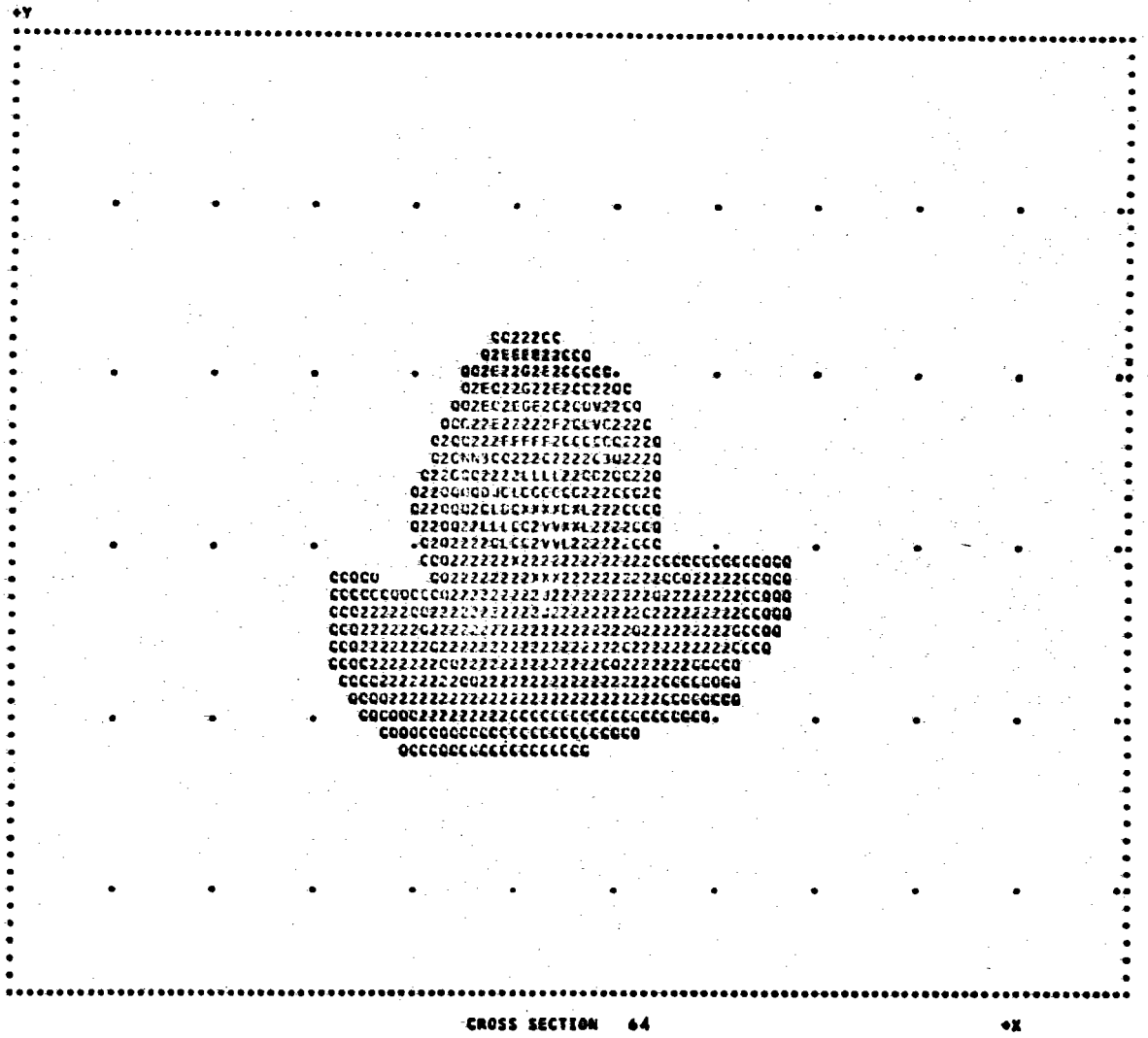
CROSS SECTION 60

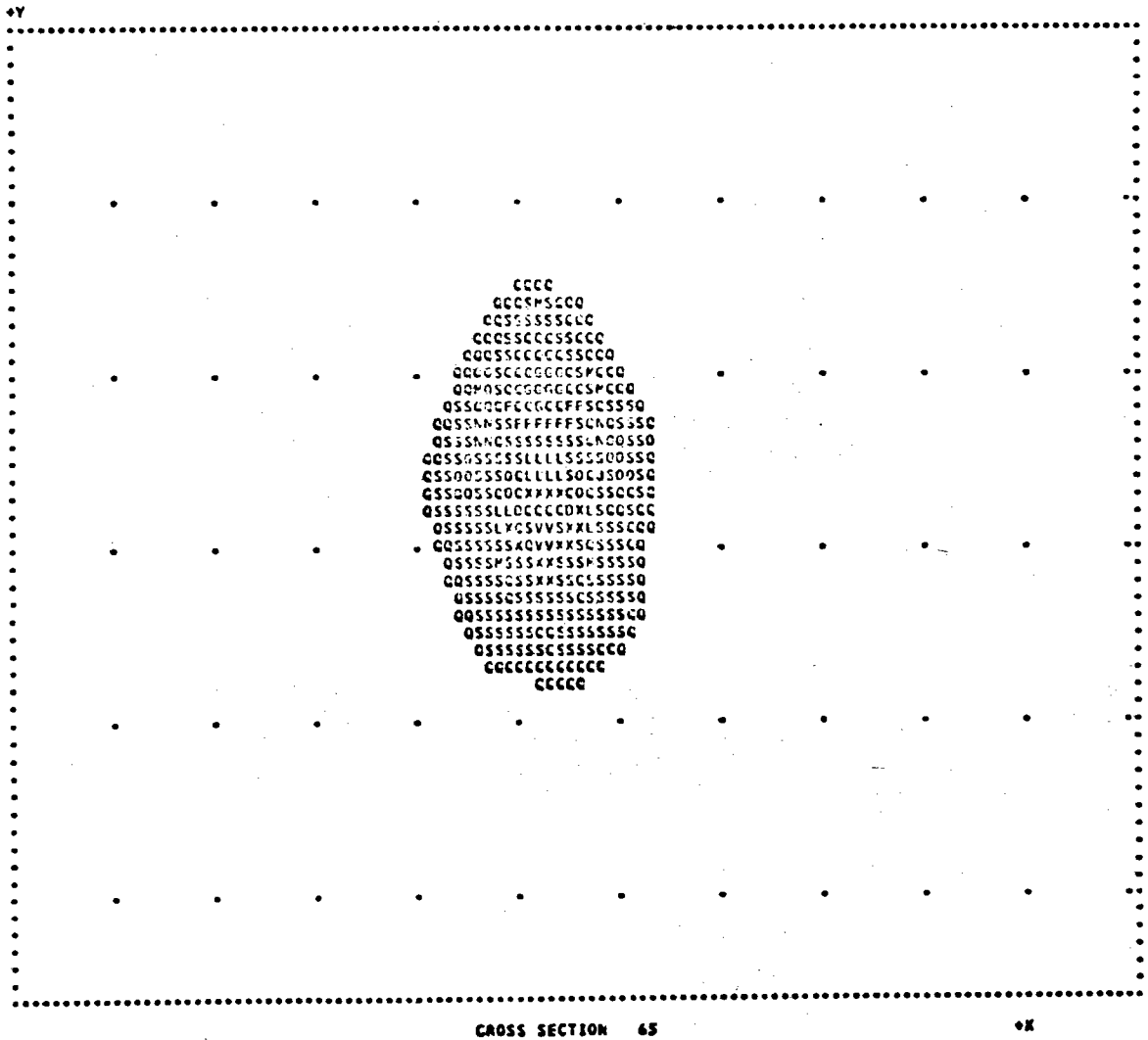


CROSS SECTION 61

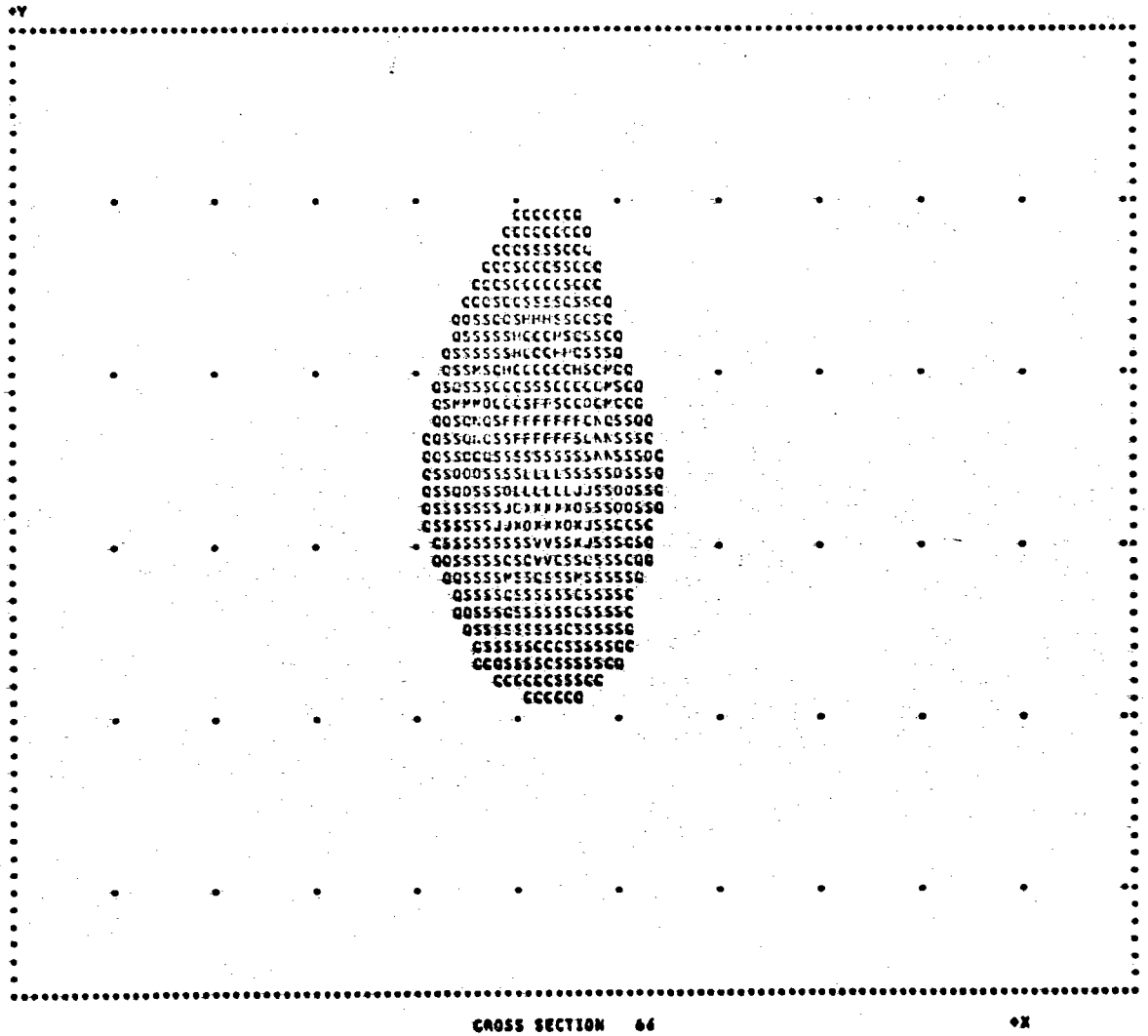

+Y
CROSS SECTION 62
+X

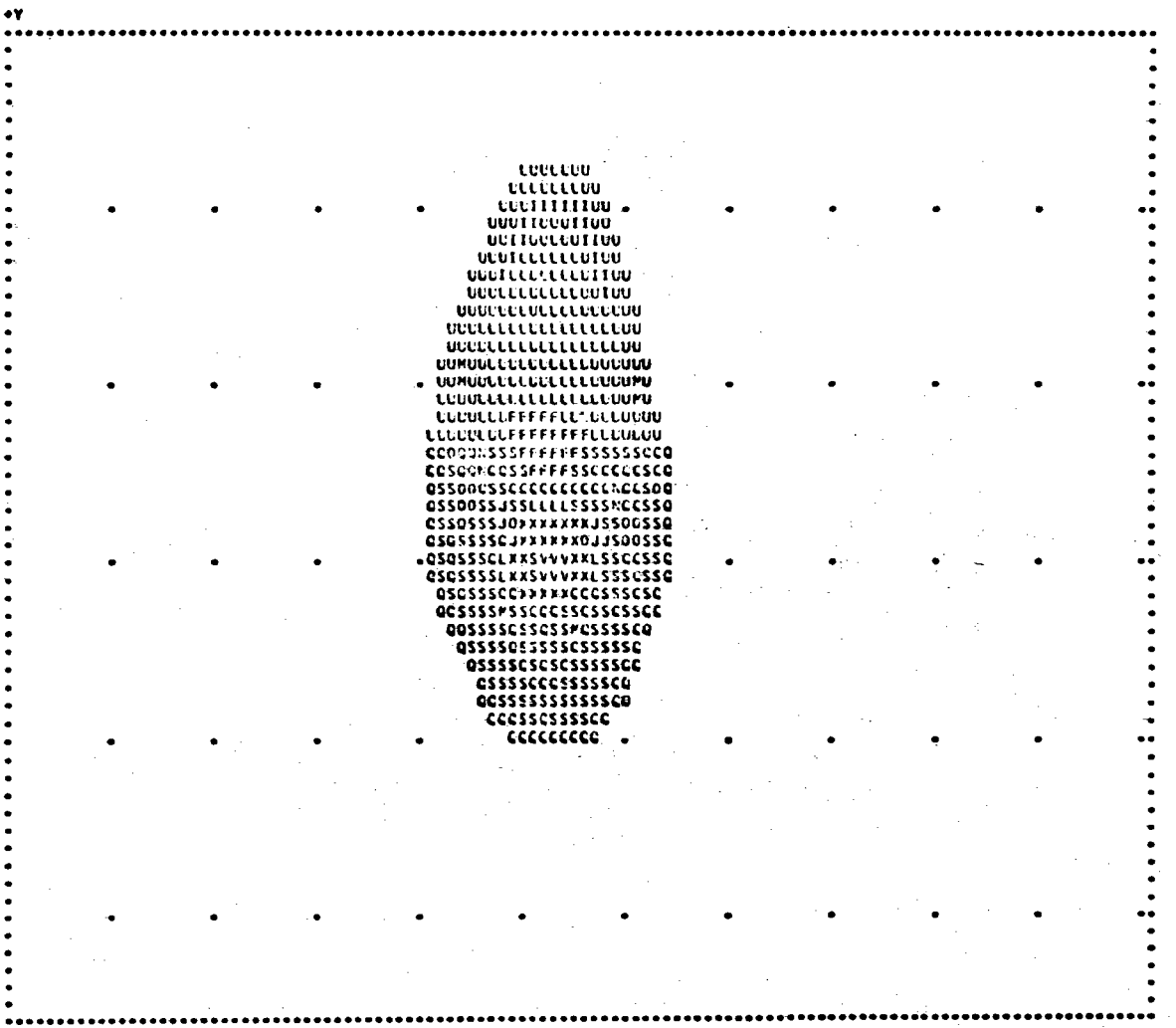

+Y
CROSS SECTION 63
+X

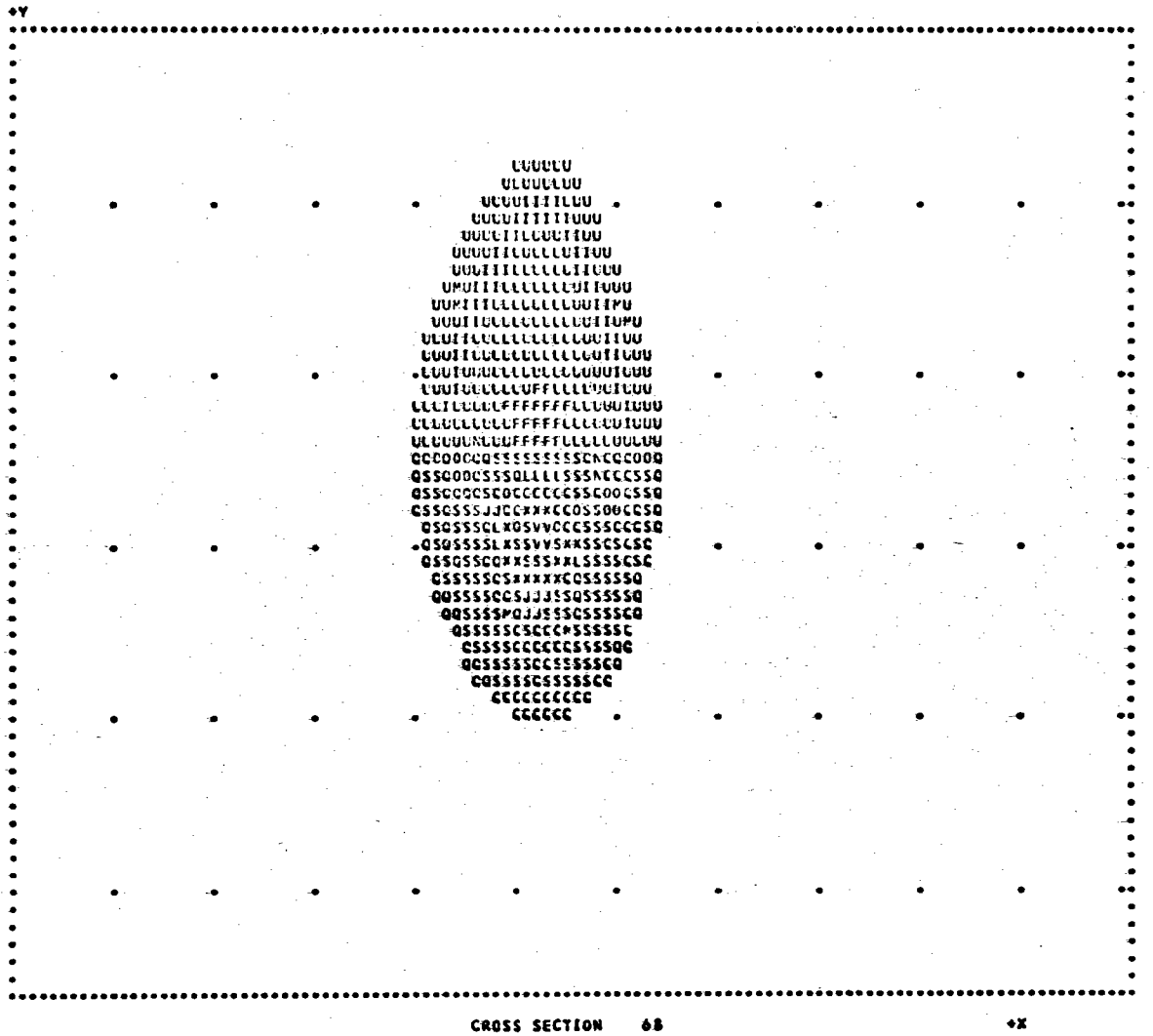

+Y
CROSS SECTION 64
+X

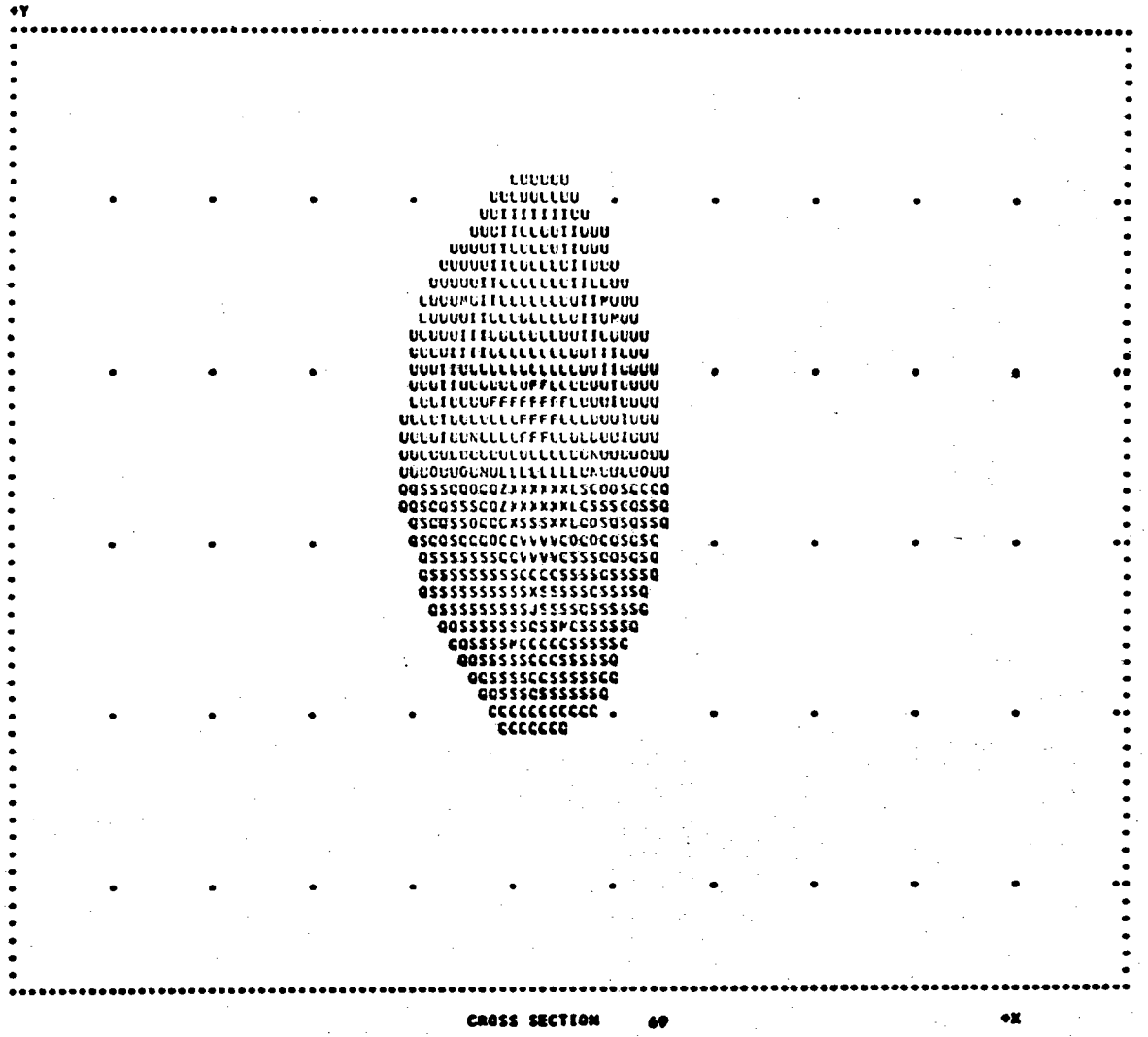


CROSS SECTION 65

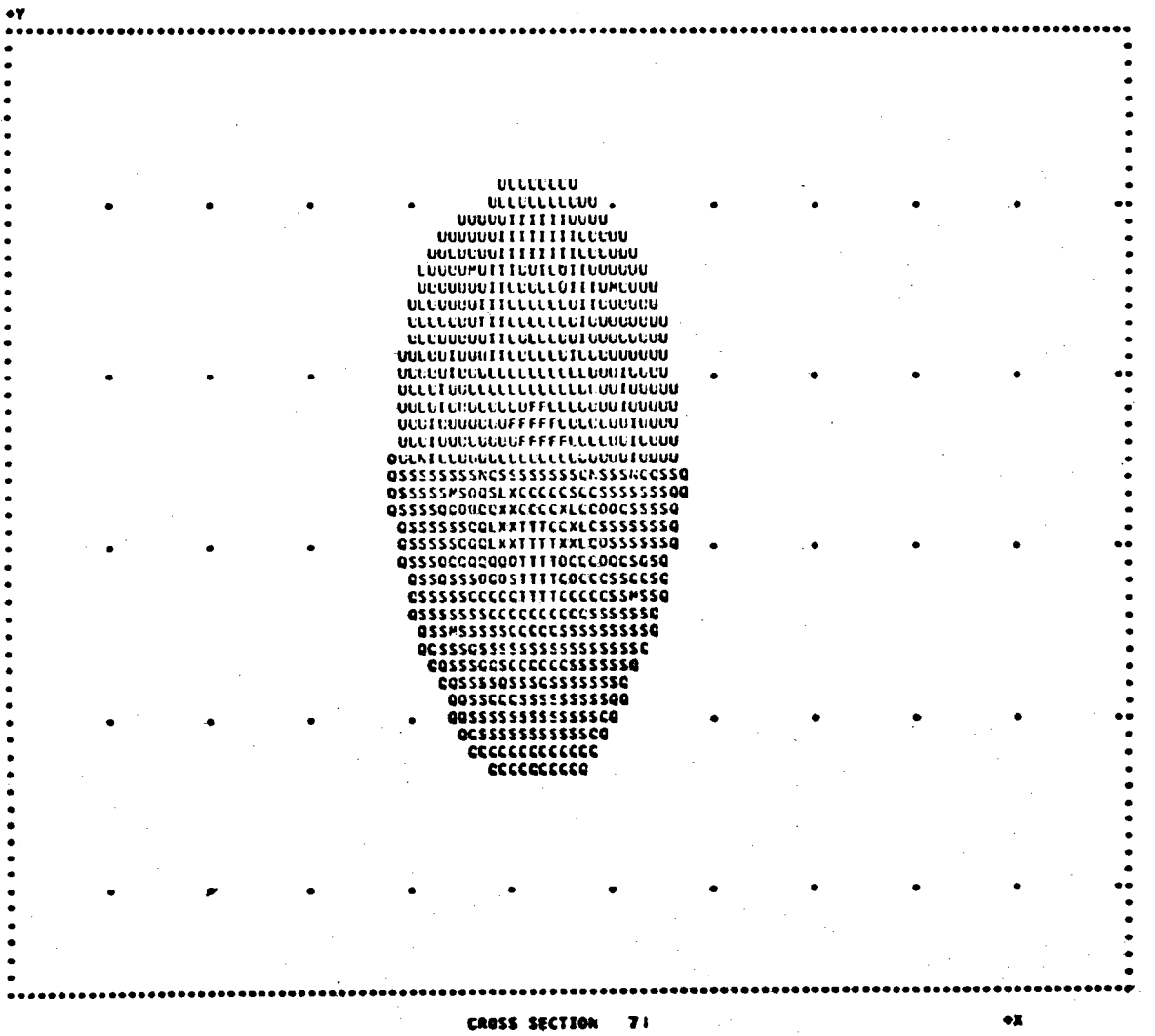

+Y
CROSS SECTION 66
+X


+Y
CROSS SECTION 67
+X

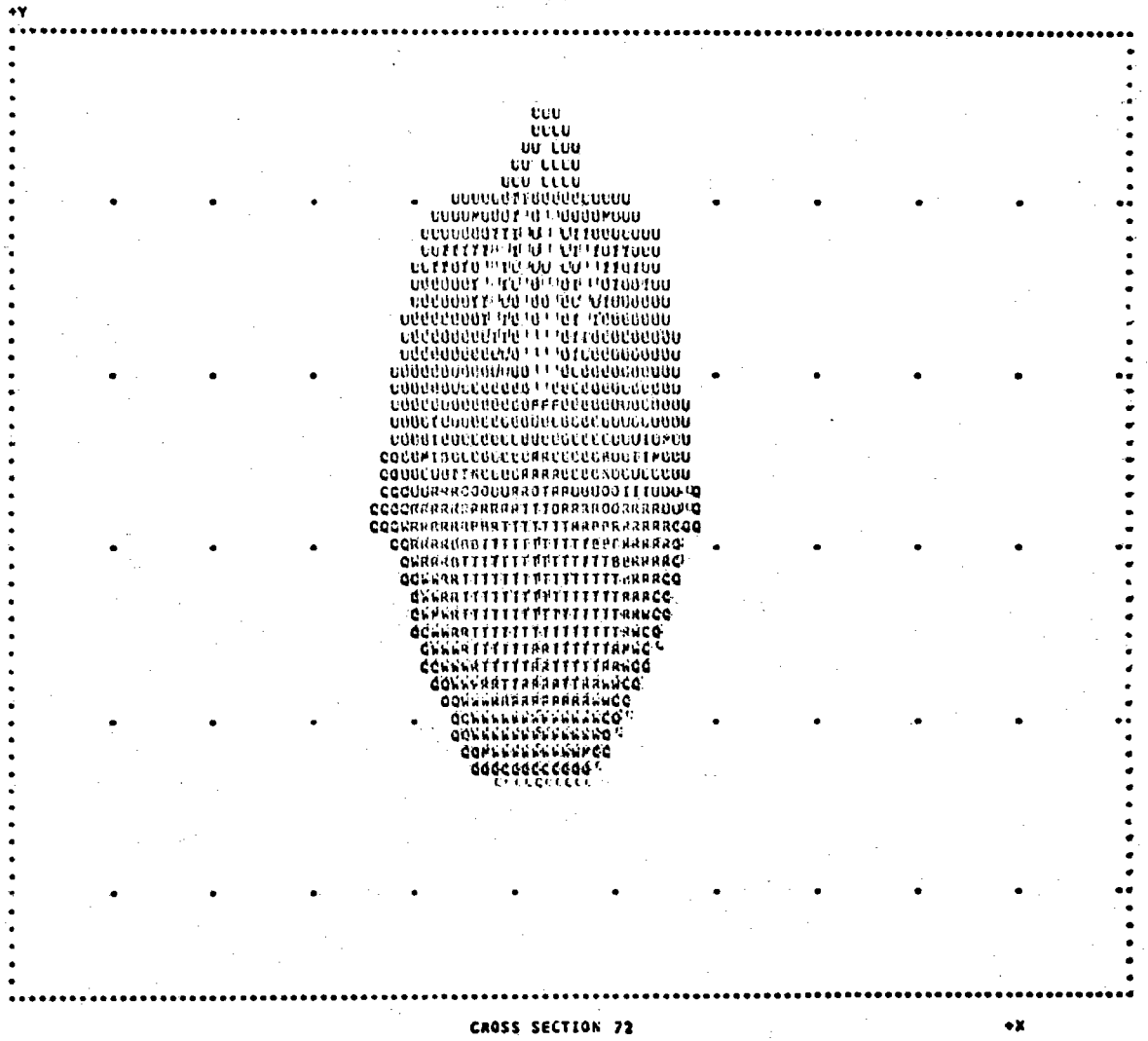

+Y
CROSS SECTION 68
+X

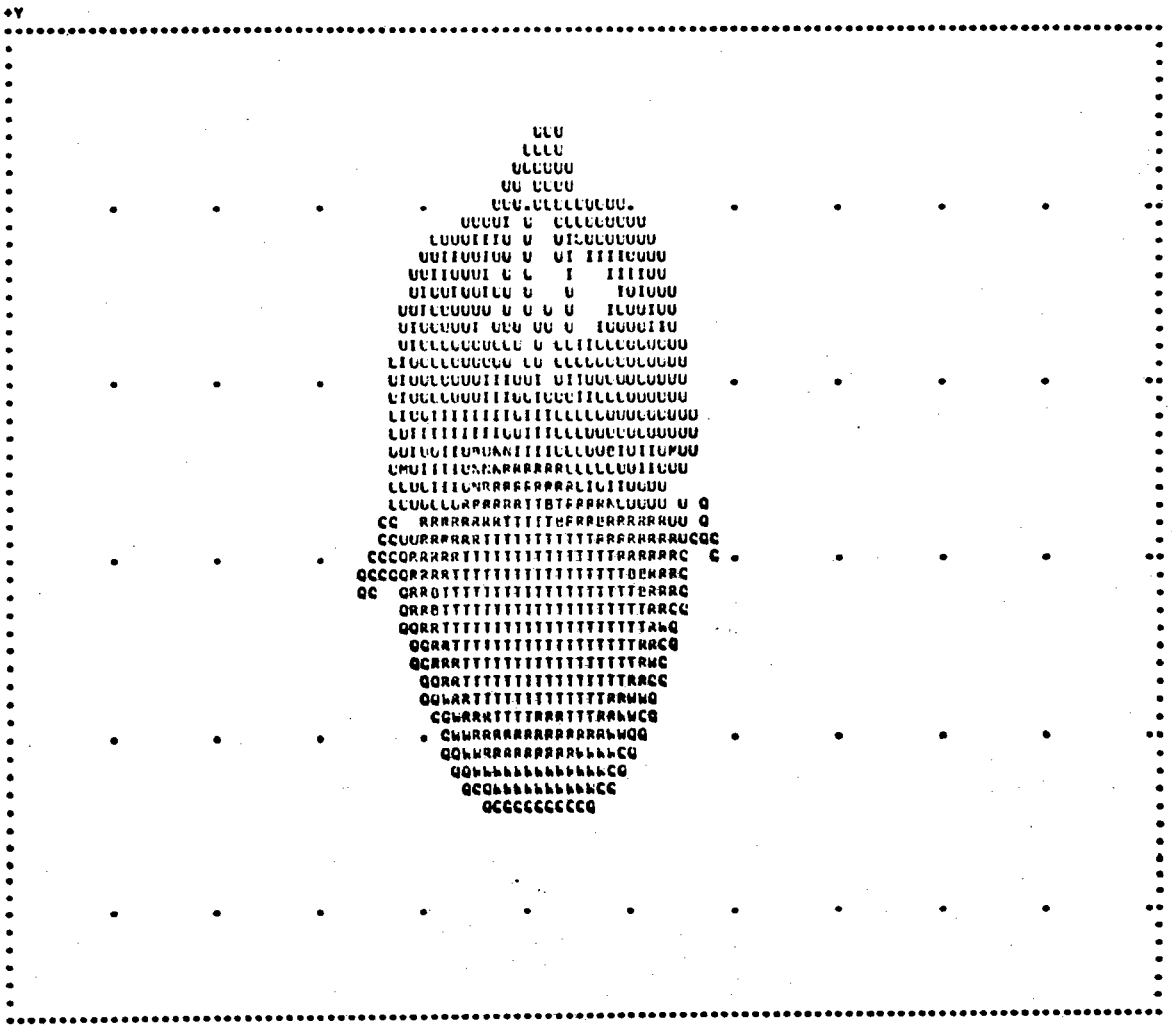

+Y
+X
CROSS SECTION 69

CROSS SECTION TO

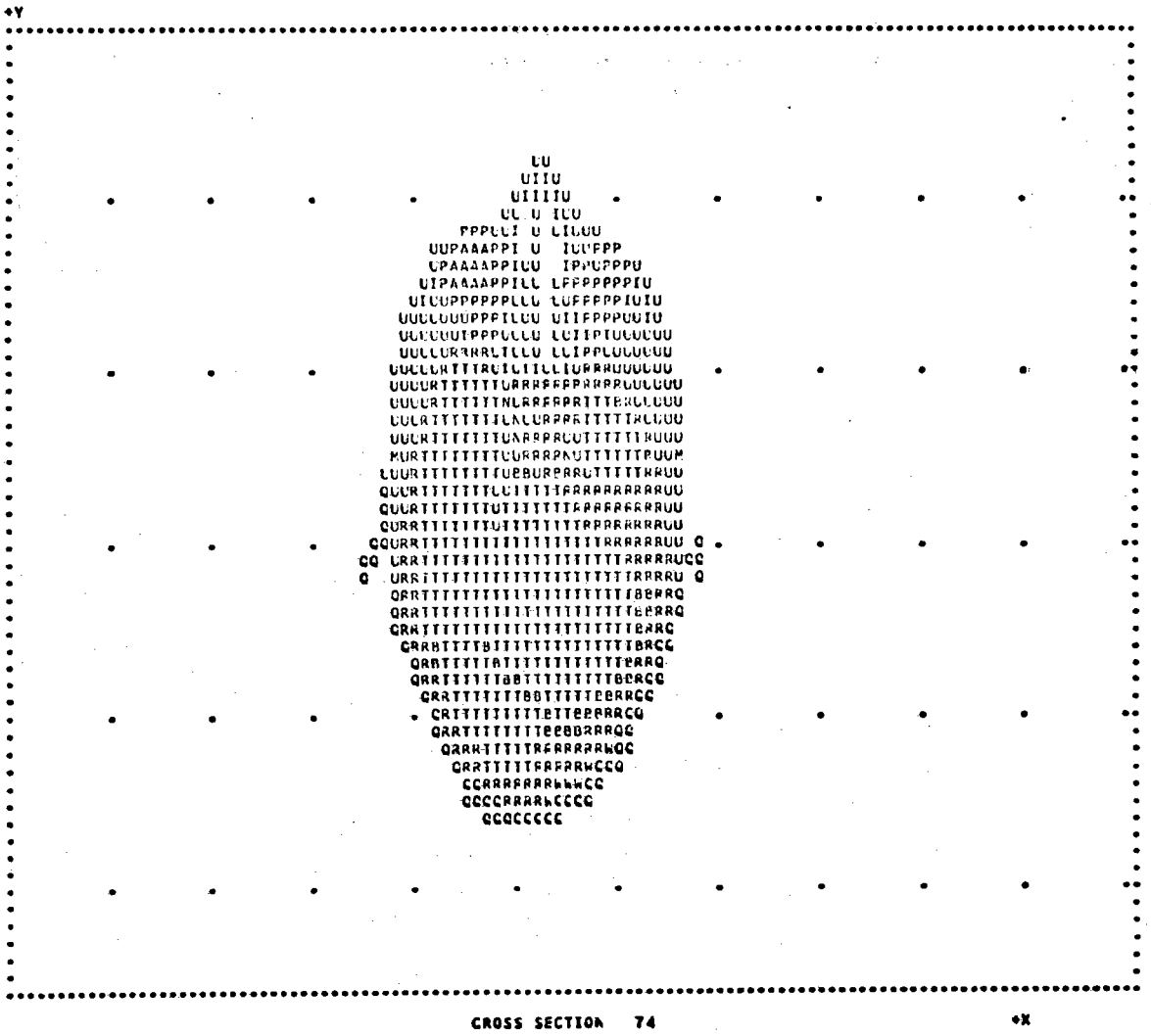


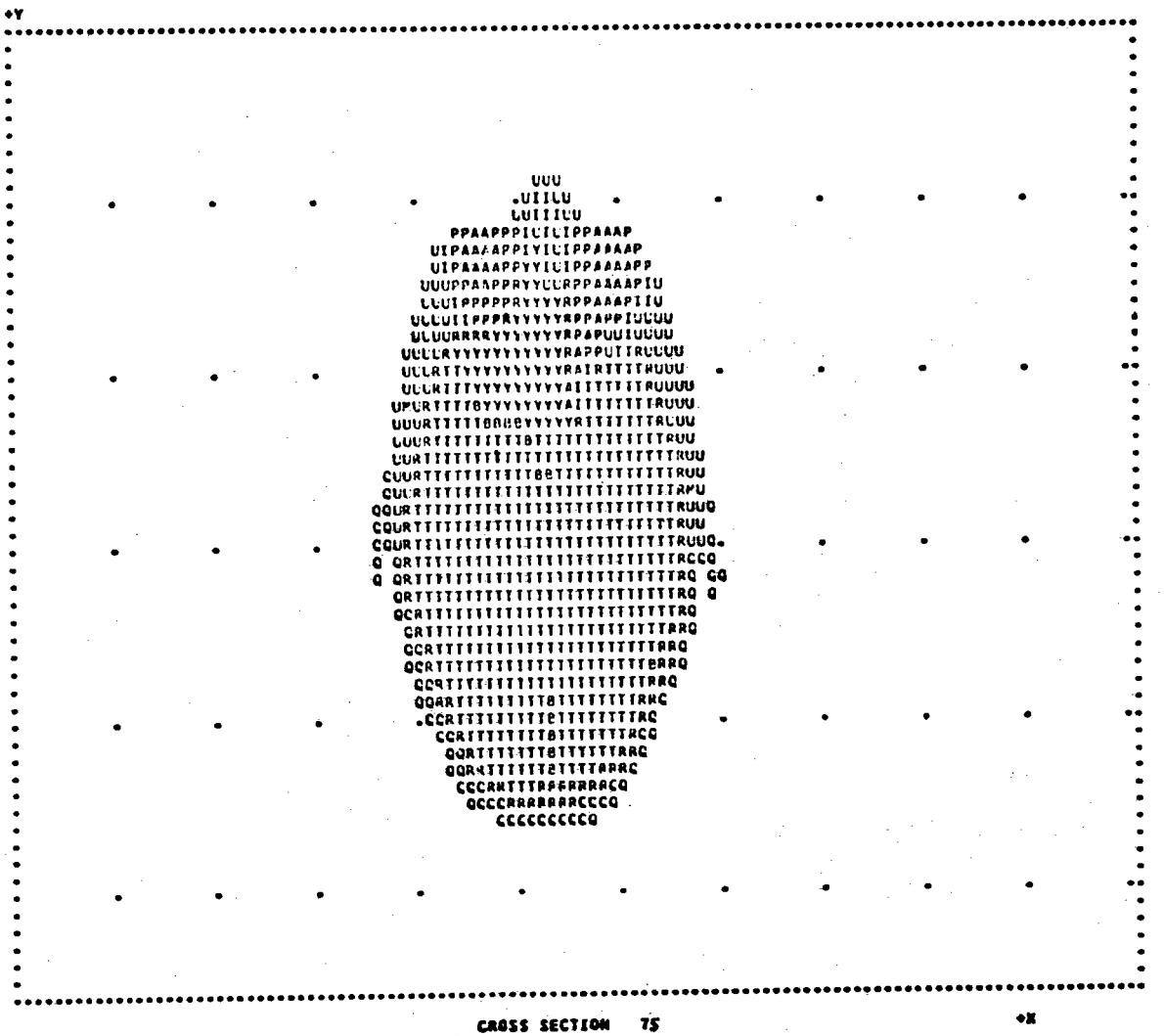
CROSS SECTION 71

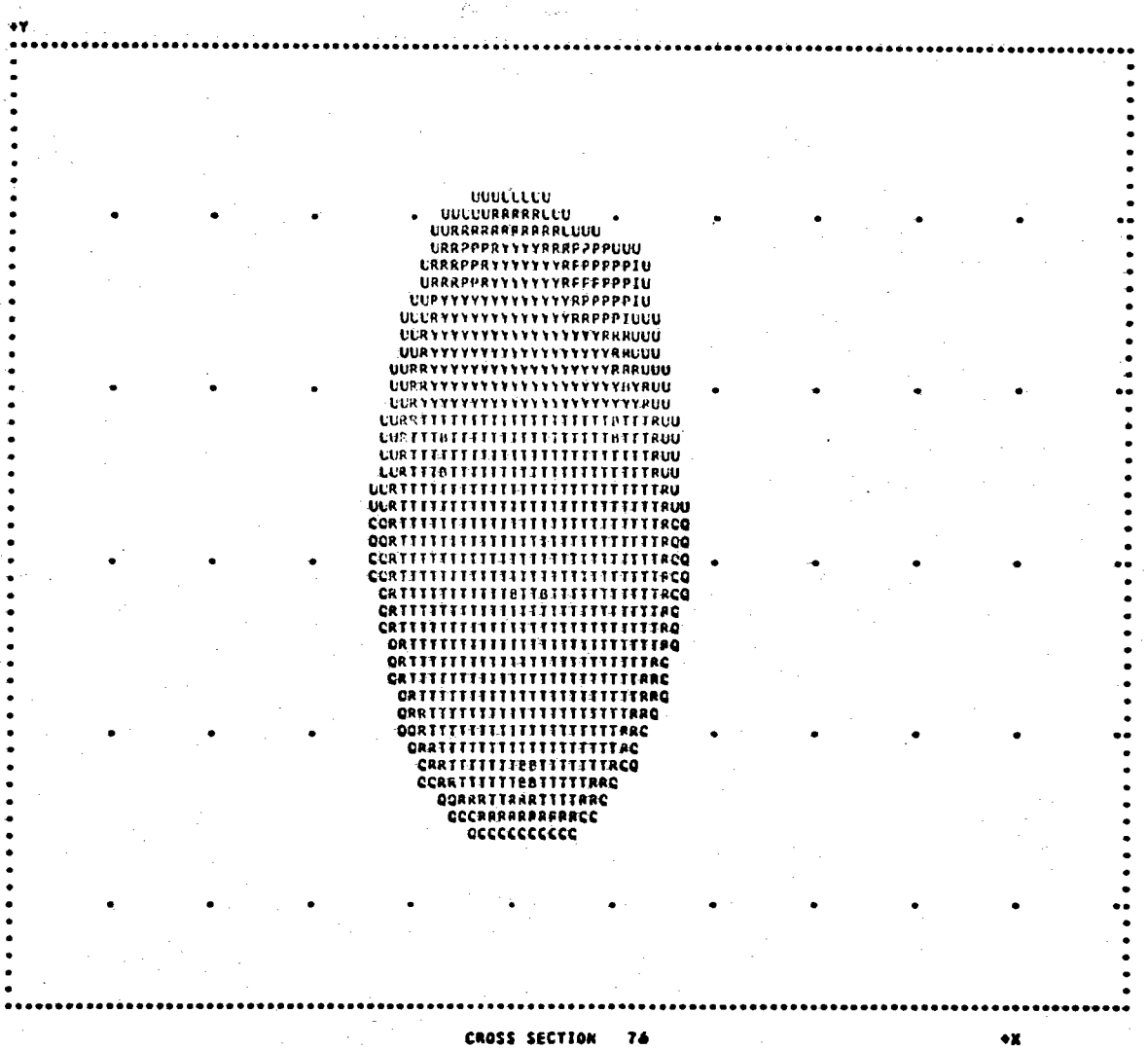

+Y
CROSS SECTION 72
+X

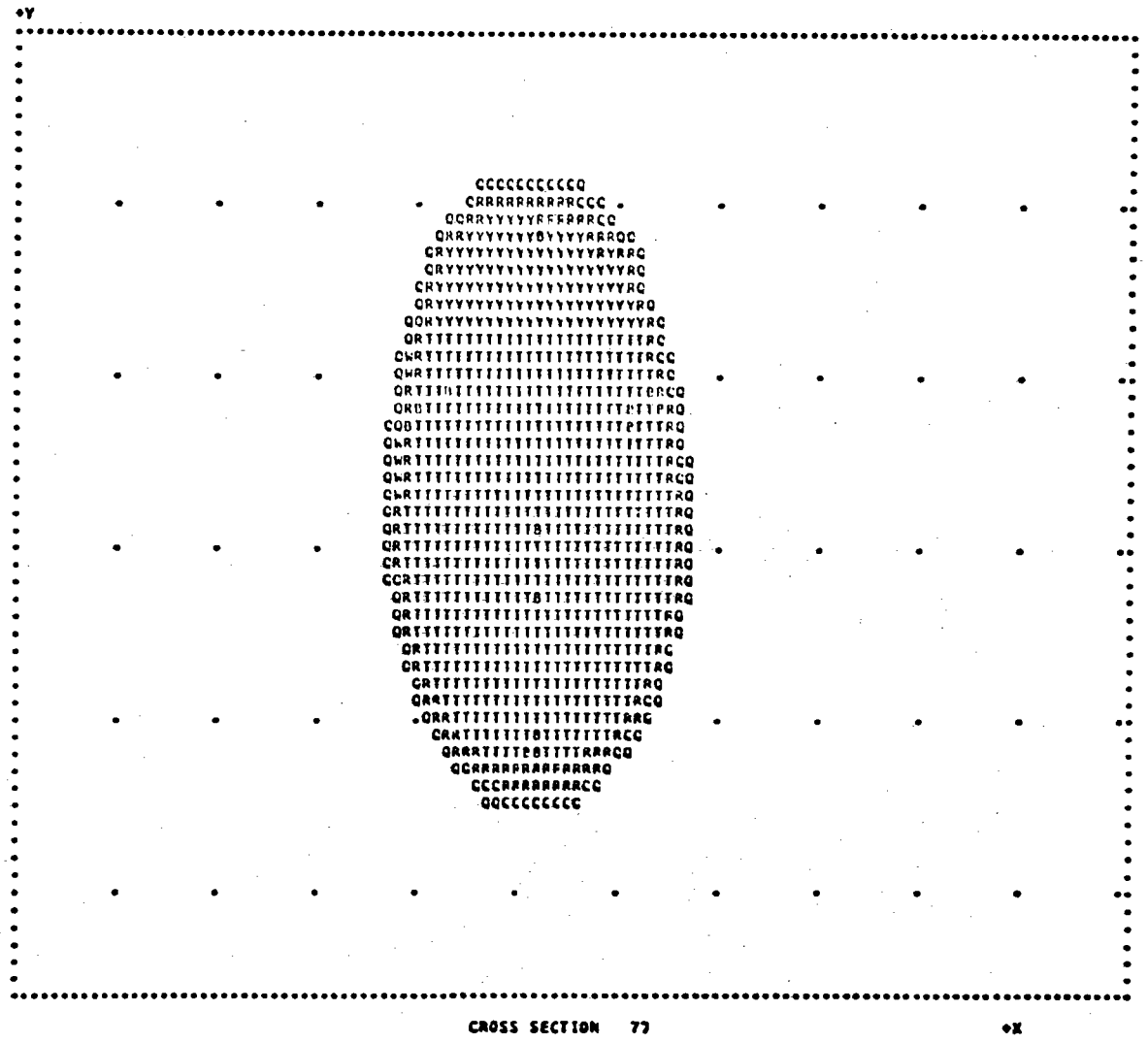


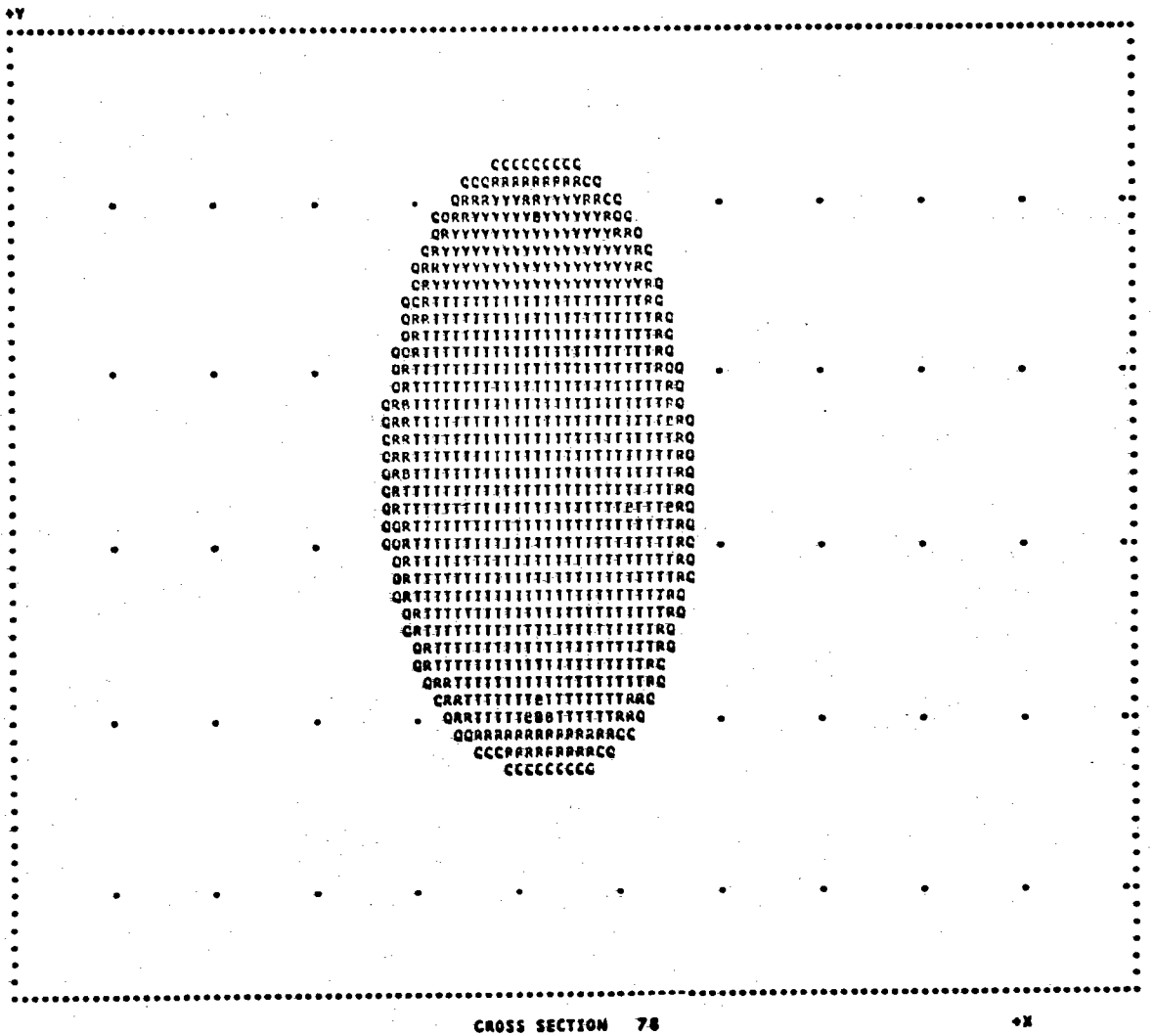
CROSS SECTION 73


+Y
+X
CROSS SECTION 74

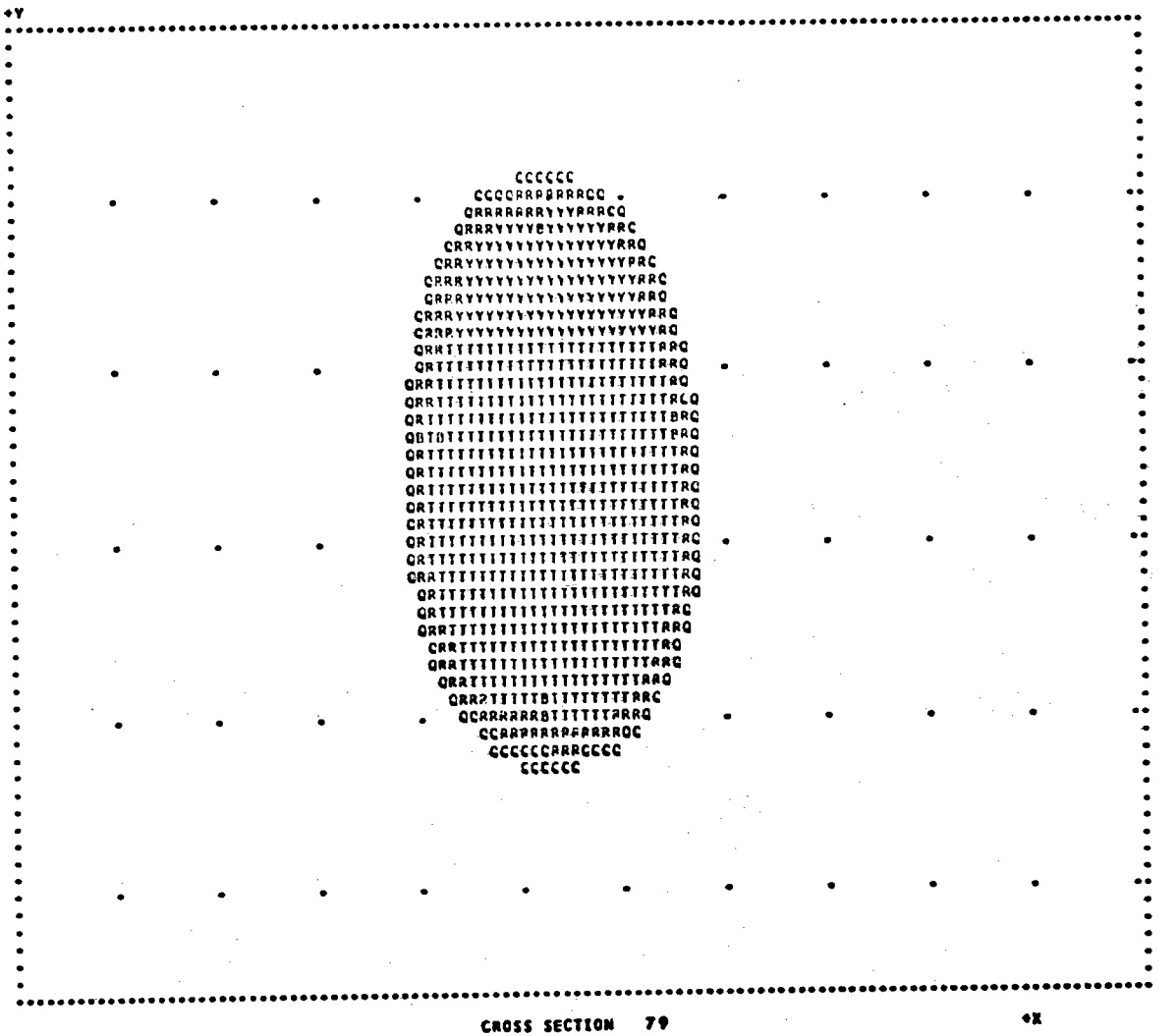

+Y
+X
CROSS SECTION 75

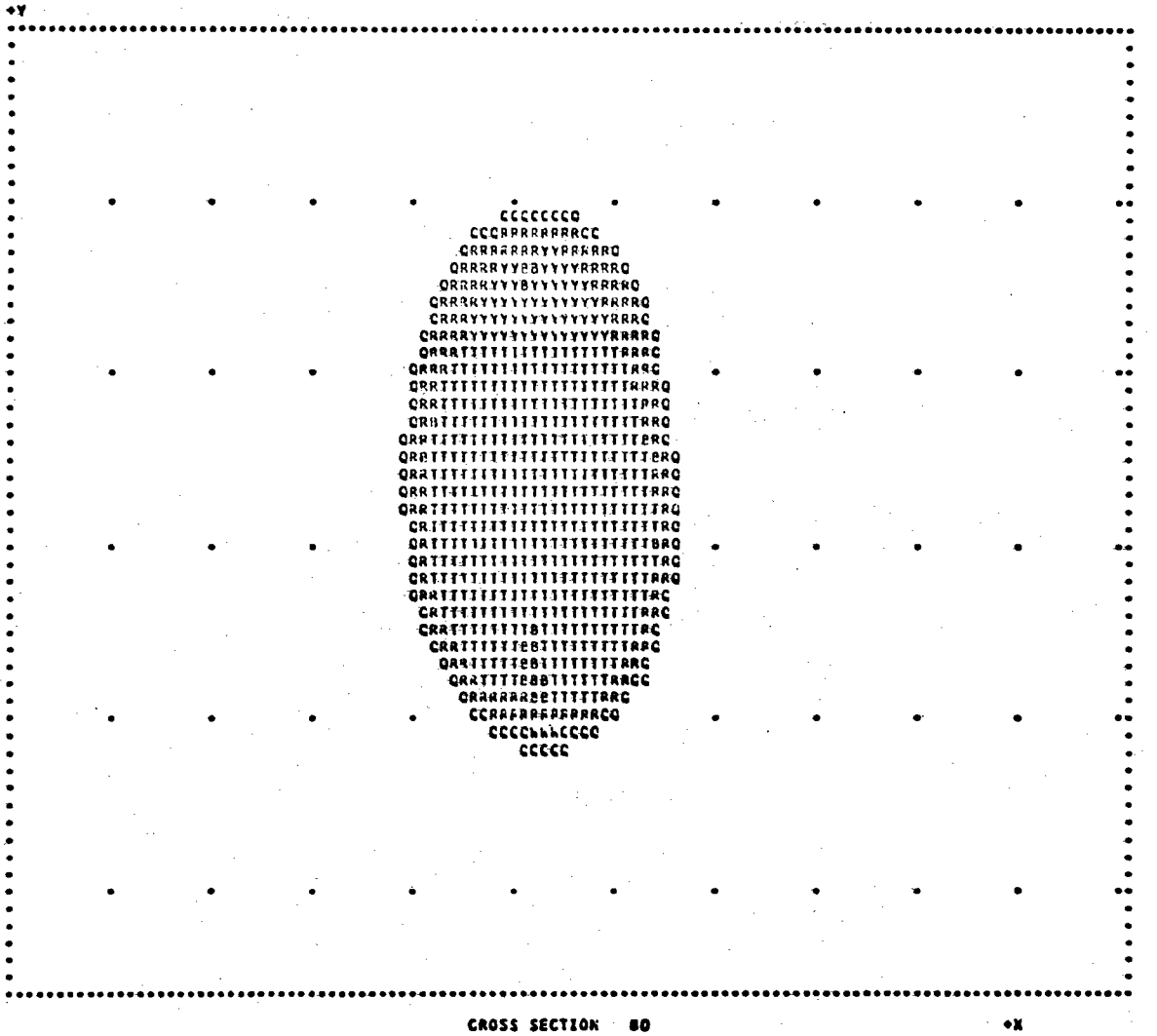

+Y
+X
CROSS SECTION 76

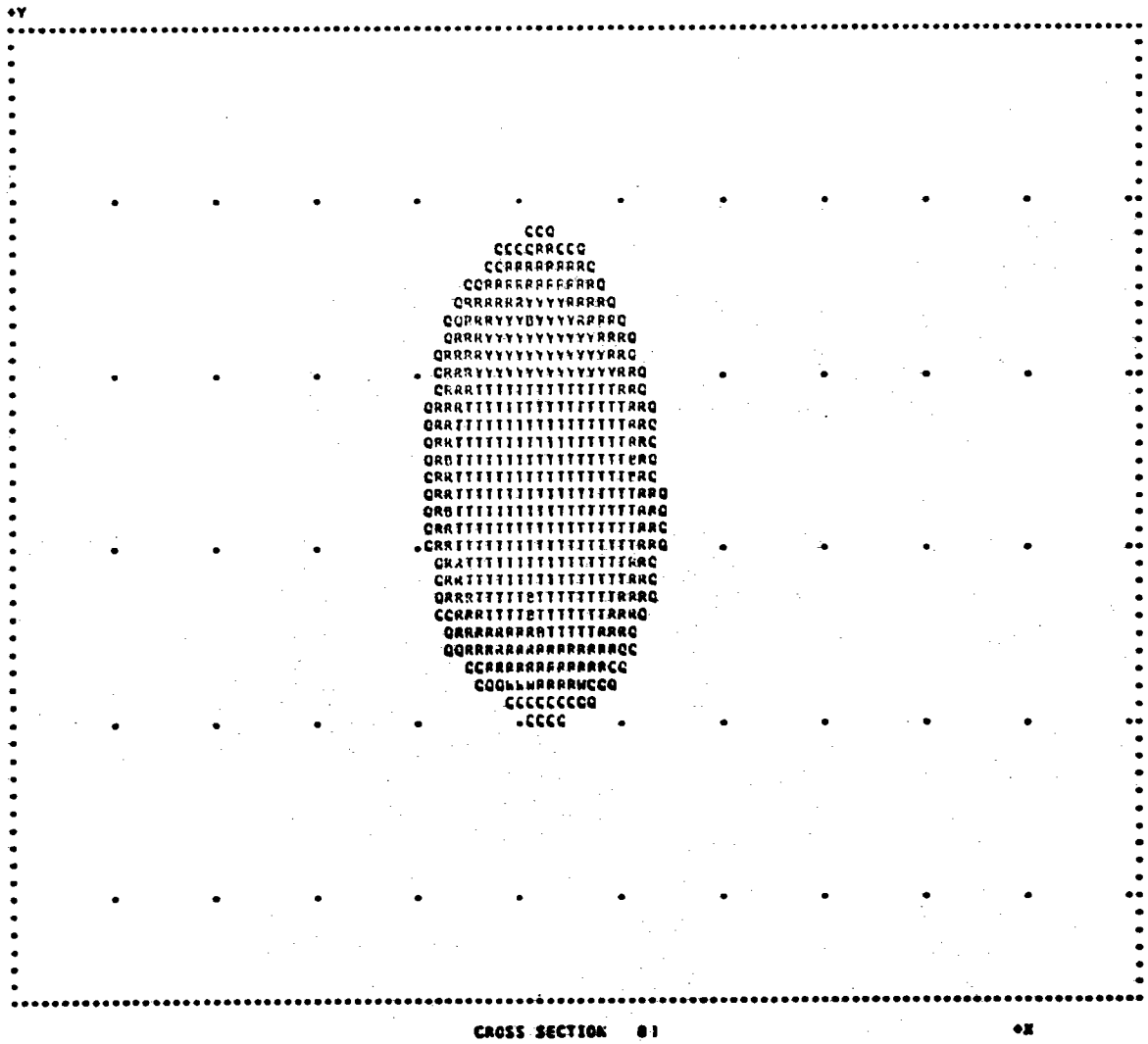

+Y
CROSS SECTION 77
+X

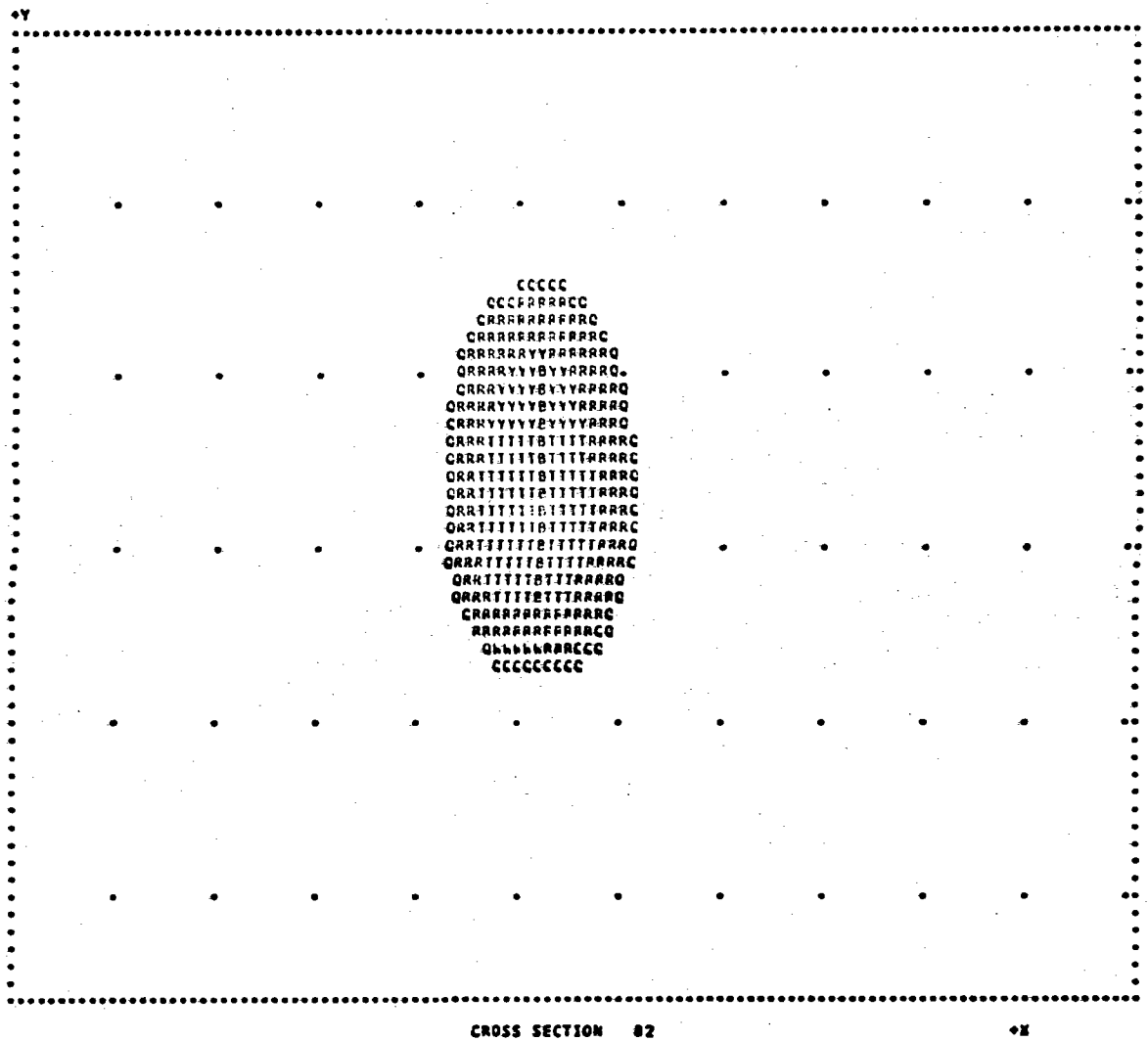


CROSS SECTION 78

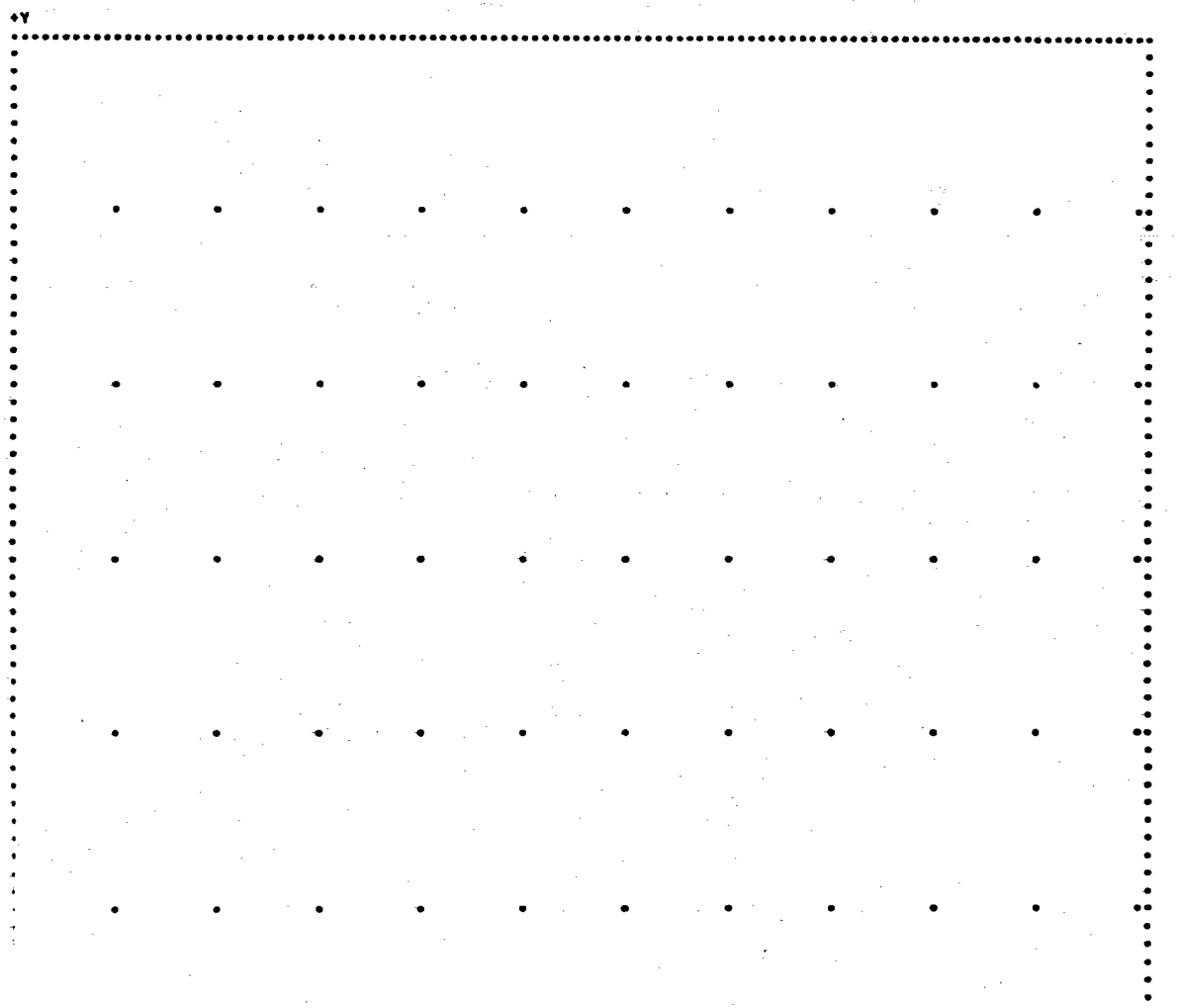

+Y
CCCCCC
CCCCRRRRRRCC
CRRRRRRRYYYRRRCC
CRRRYYYYBYYYYYYRRC
CRRYYYYYYYYYYYYYYRRC
CRRYYYYYYYYYYYYYYYRRC
CRRRYYYYYYYYYYYYYYYYRRC
CRRRYYYYYYYYYYYYYYYYRRC
CRRRYYYYYYYYYYYYYYYYYRRC
CRRRYYYYYYYYYYYYYYYYYRRC
CRRTTTTTTTTTTTTTTTTTTTTTRRC
CRTTTTTTTTTTTTTTTTTTTTTTRRC
CRRTTTTTTTTTTTTTTTTTTTTTTRC
CRRTTTTTTTTTTTTTTTTTTTTTTRLC
CRTTTTTTTTTTTTTTTTTTTTTTTBRC
CBTDTTTTTTTTTTTTTTTTTTTTTTPRC
CRTTTTTTTTTTTTTTTTTTTTTTTTRC
CRTTTTTTTTTTTTTTTTTTTTTTTTRC
CRTTTTTTTTTTTTTTTTTTTTTTTTRC
CRTTTTTTTTTTTTTTTTTTTTTTTTRC
CRTTTTTTTTTTTTTTTTTTTTTTTTRC
CRTTTTTTTTTTTTTTTTTTTTTTTTRC
CRTTTTTTTTTTTTTTTTTTTTTTTTRC
CRRTTTTTTTTTTTTTTTTTTTTTTTRC
CRTTTTTTTTTTTTTTTTTTTTTTTRC
CRRTTTTTTTTTTTTTTTTTTTTTTRC
CRTTTTTTTTTTTTTTTTTTTTTTRC
CRRTTTTTTTTTTTTTTTTTTTTTRRC
CRRTTTTTTTTTTTTTTTTTTTTRRC
CRRTTTTTTTTTTTTTTTTTTTRRC
CRRTTTTTTTTTTTTTTTTTTRRC
CRRTTTTTTTTTTTTTTTTTRRC
CRRTTTTTTTTTTTTTTTTRRC
CRRTTTTTTTTTTTTTTTRRC
CCRRRRRRBTTTTTTRRC
CCRRRRRRRRRRRCC
CCCCCCRRRCCCC
CCCCCC
CROSS SECTION 79
+X


+Y
CROSS SECTION 80
+X


+Y
CCQ
CCCCRRCCQ
CCRRRRRRRC
CQRRRRRRRRRQ
QRRRRRYYYYRRRQ
QQPRRYYYYYYYRRRQ
QRRRYYYYYYYYYYRRRQ
QRRRRYYYYYYYYYYRRQ
CRRRYYYYYYYYYYYYRRQ
CRRRTTTTTTTTTTTTTRRQ
QRRRTTTTTTTTTTTTTTTRRQ
QRRTTTTTTTTTTTTTTTTRRC
QRRTTTTTTTTTTTTTTTTRRC
QRQTTTTTTTTTTTTTTTTRRQ
CRRTTTTTTTTTTTTTTTTRRC
QRRTTTTTTTTTTTTTTTTTRRQ
QRQTTTTTTTTTTTTTTTTTRRQ
QRRTTTTTTTTTTTTTTTTTTRRC
CRRTTTTTTTTTTTTTTTTTTRRQ
CRRTTTTTTTTTTTTTTTTTRRC
CRRTTTTTTTTTTTTTTTTTRRC
QRRRTTTTTQTTTTTTTTRRRQ
CCRRRTTTTQTTTTTTTRRRQ
QRRRRRRRRRTTTTTRRRQ
QQRRRRRRRRRRRRRRRQC
CCRRRRRRRRRRRRRCQ
CQQLLWRRRRWCCQ
CCCCCCCCQ
.CCCC
CROSS SECTION 81
+X


+Y
CROSS SECTION 82
+X

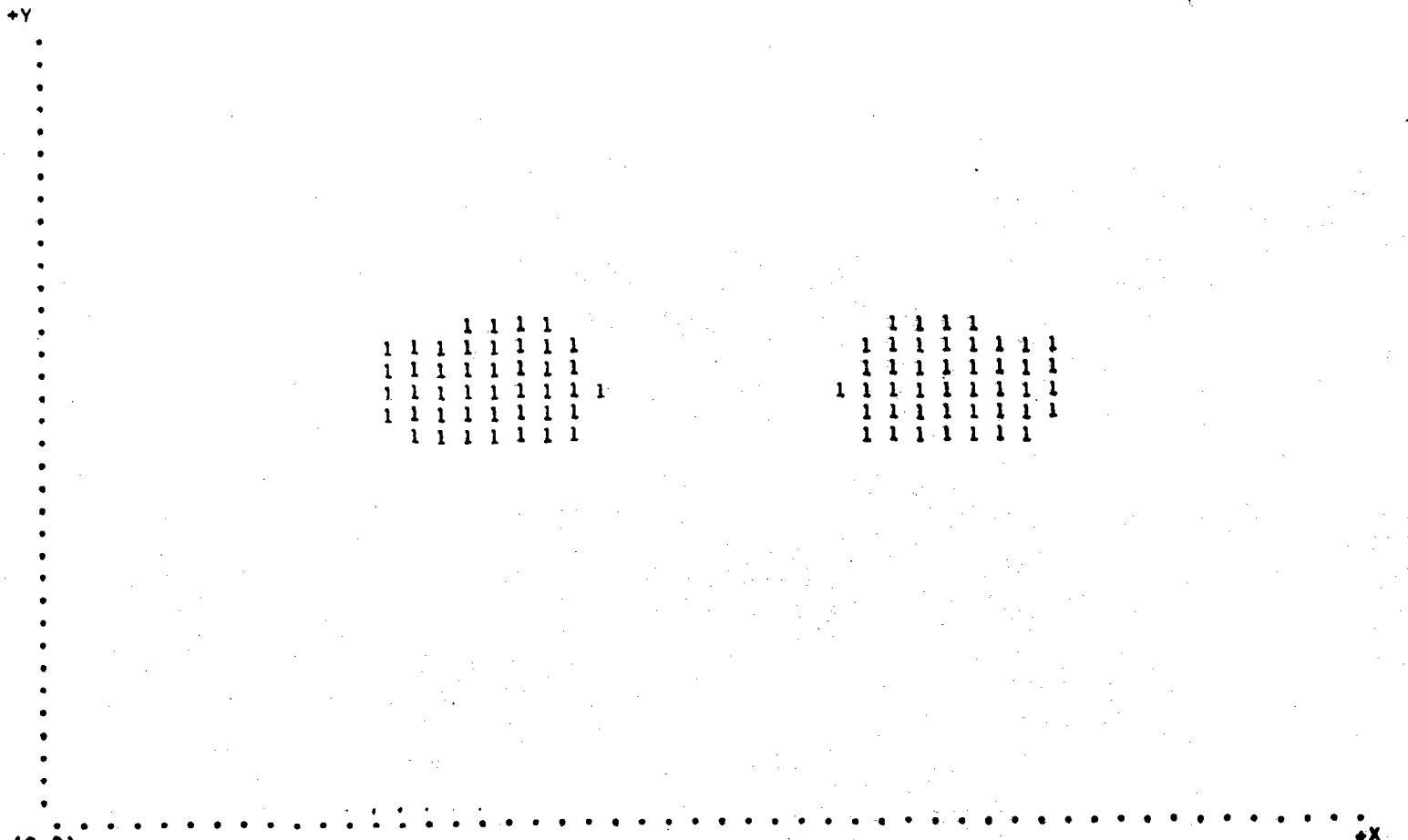
CROSS SECTION 83 (DUMMY REGION)



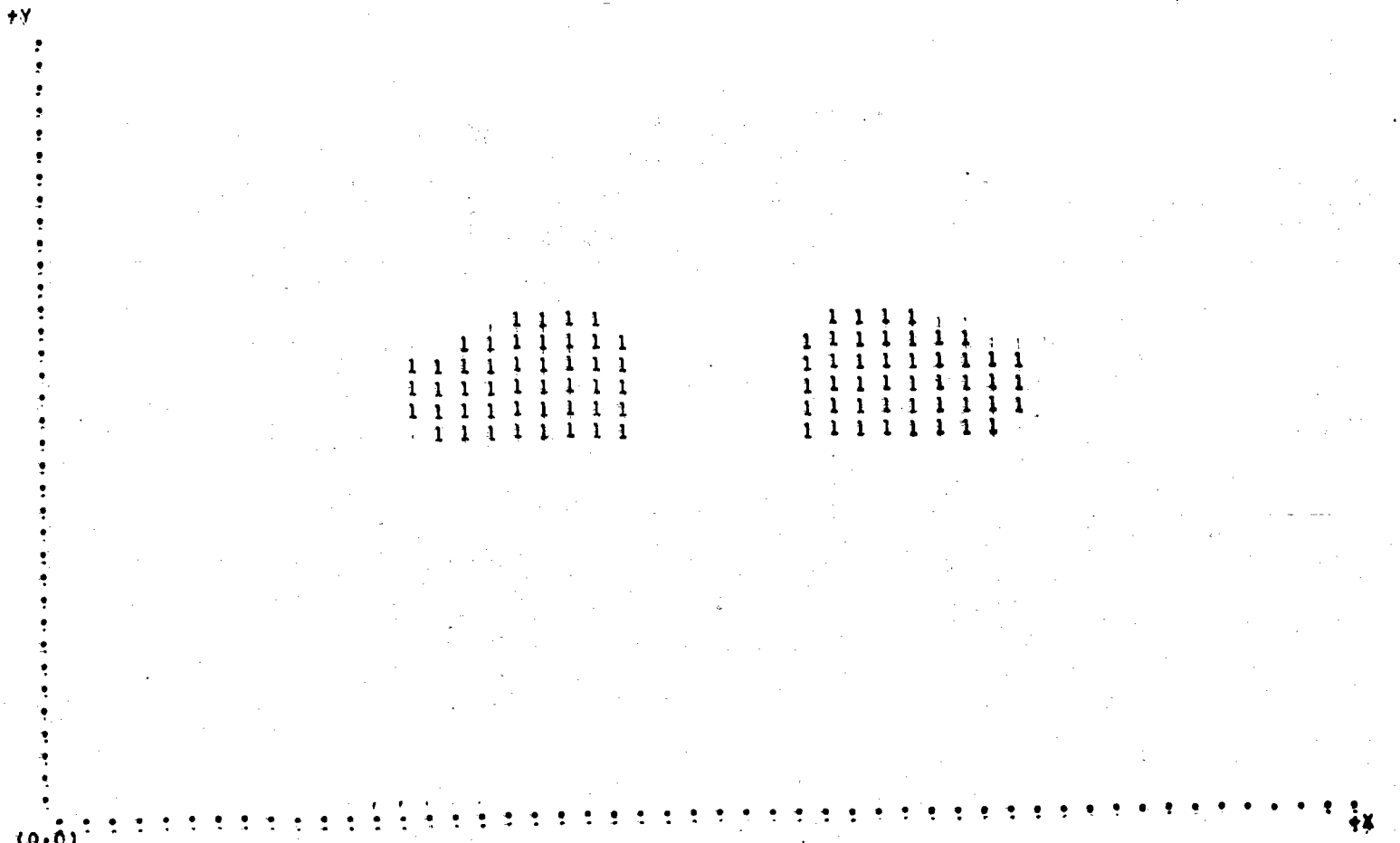
CROSS SECTION 84 (DUMMY REGION)

APPENDIX C

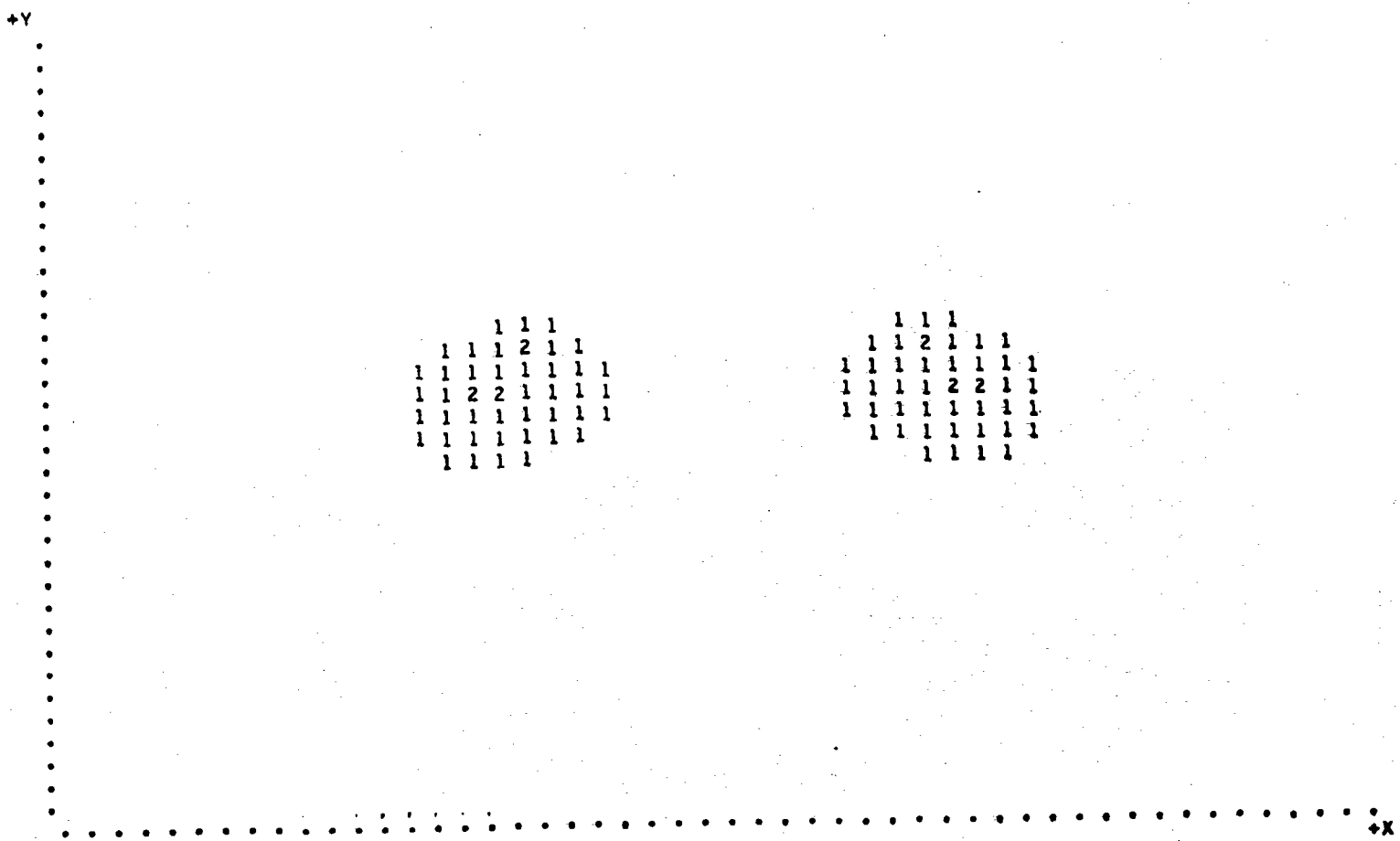
A SLICE BY SLICE COMPUTER DESCRIPTION OF THE LETHALITY MODEL



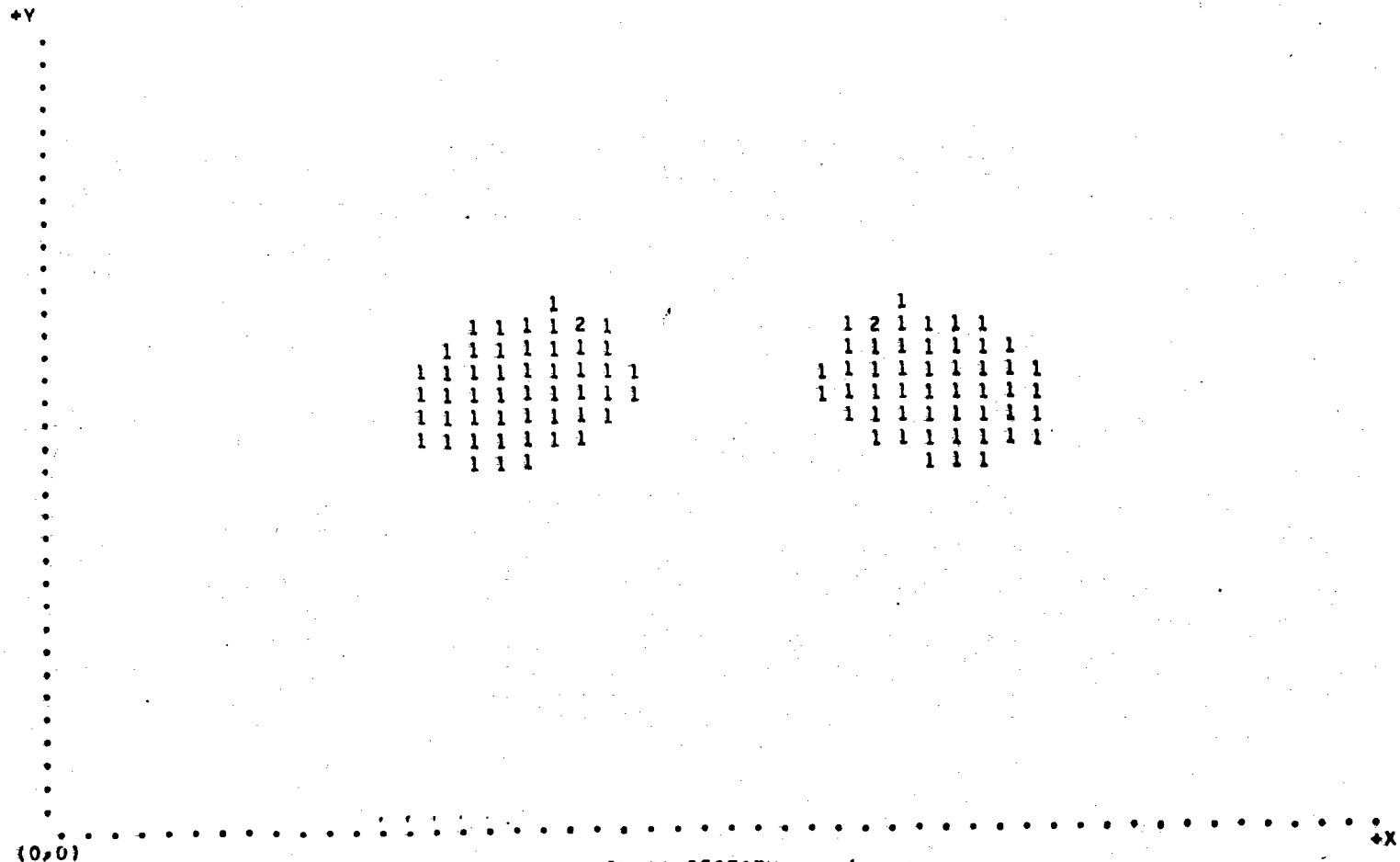
CROSS SECTION 1
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



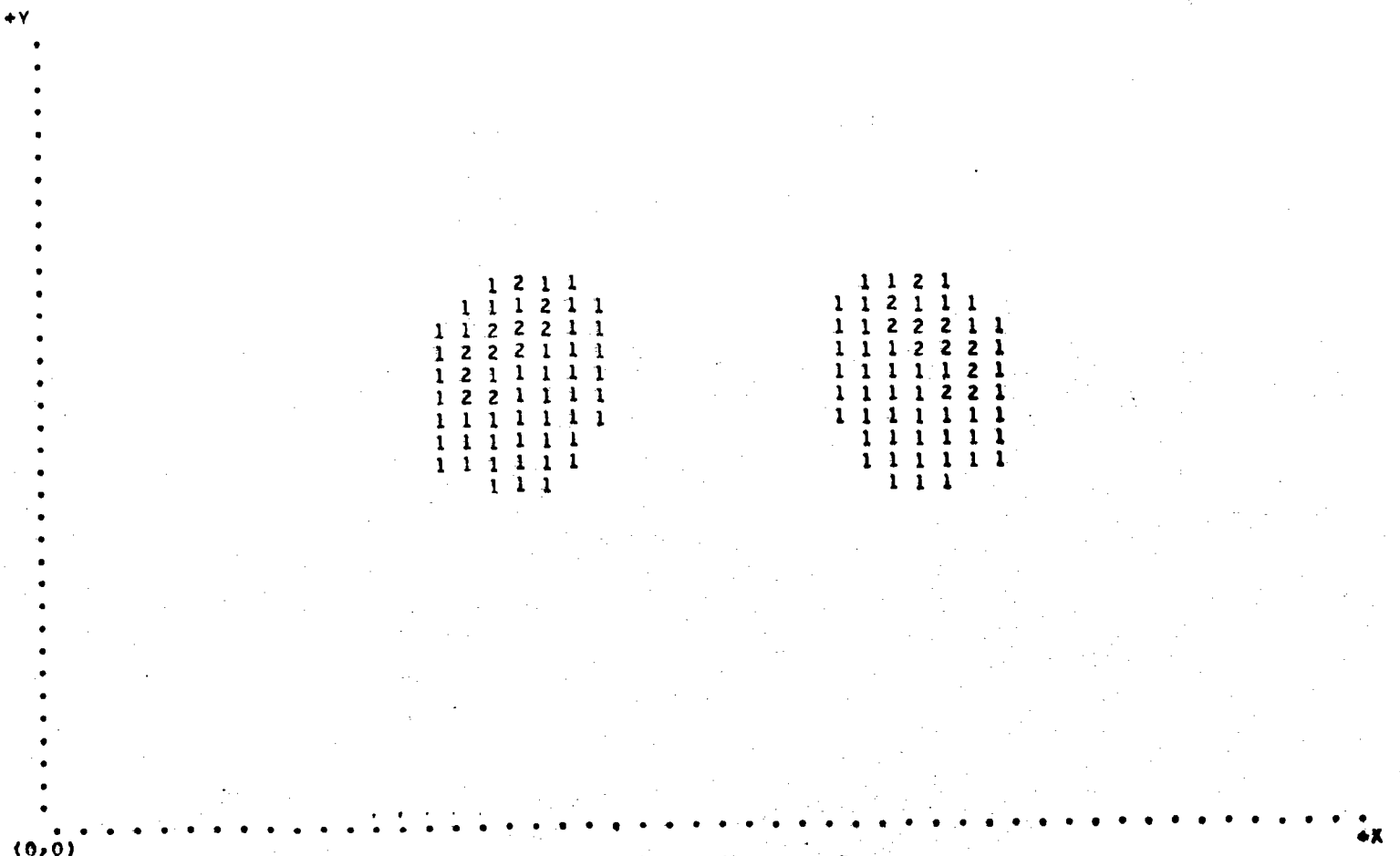
CROSS SECTION 2
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



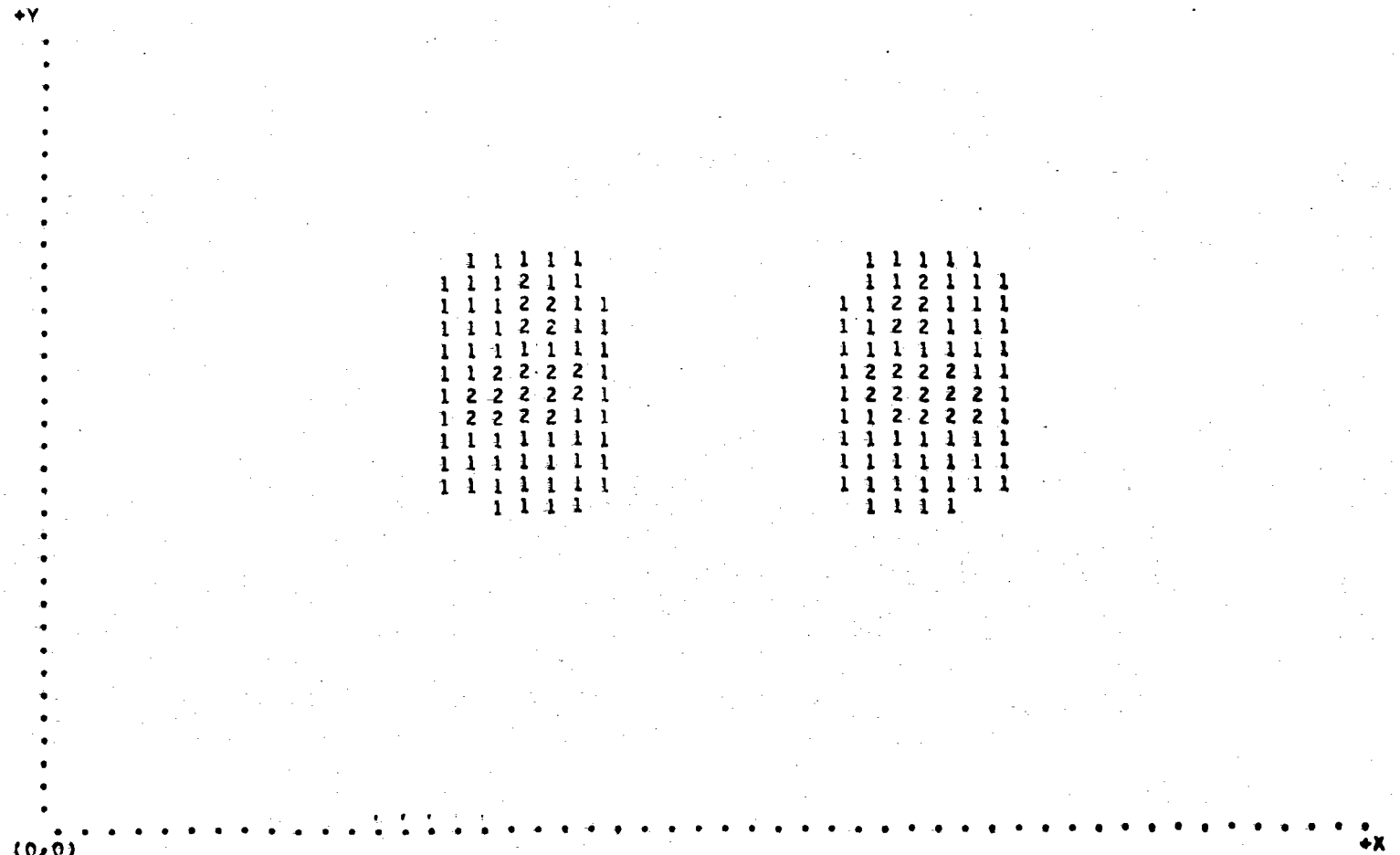
CROSS SECTION 3
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



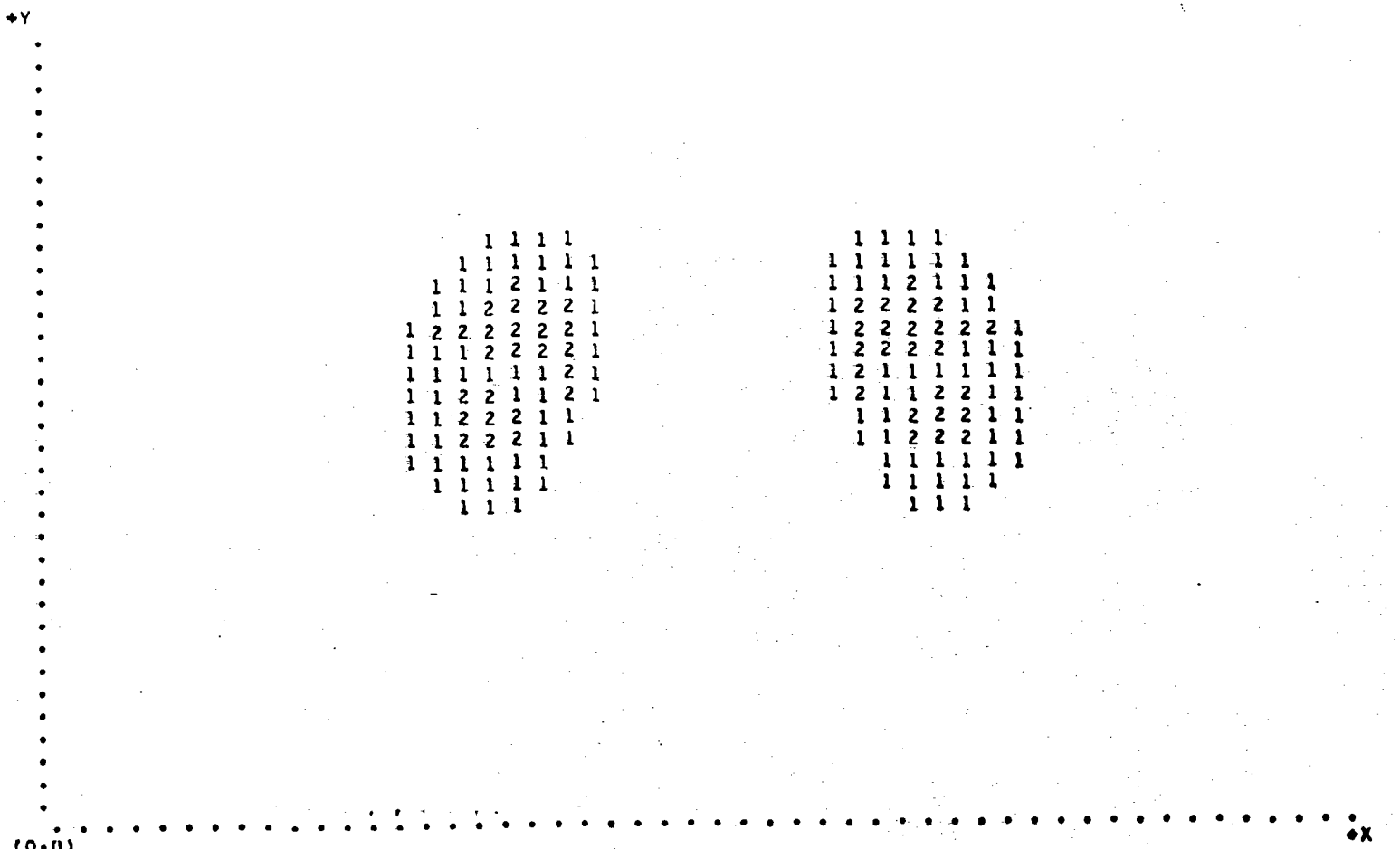
CROSS SECTION 4
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



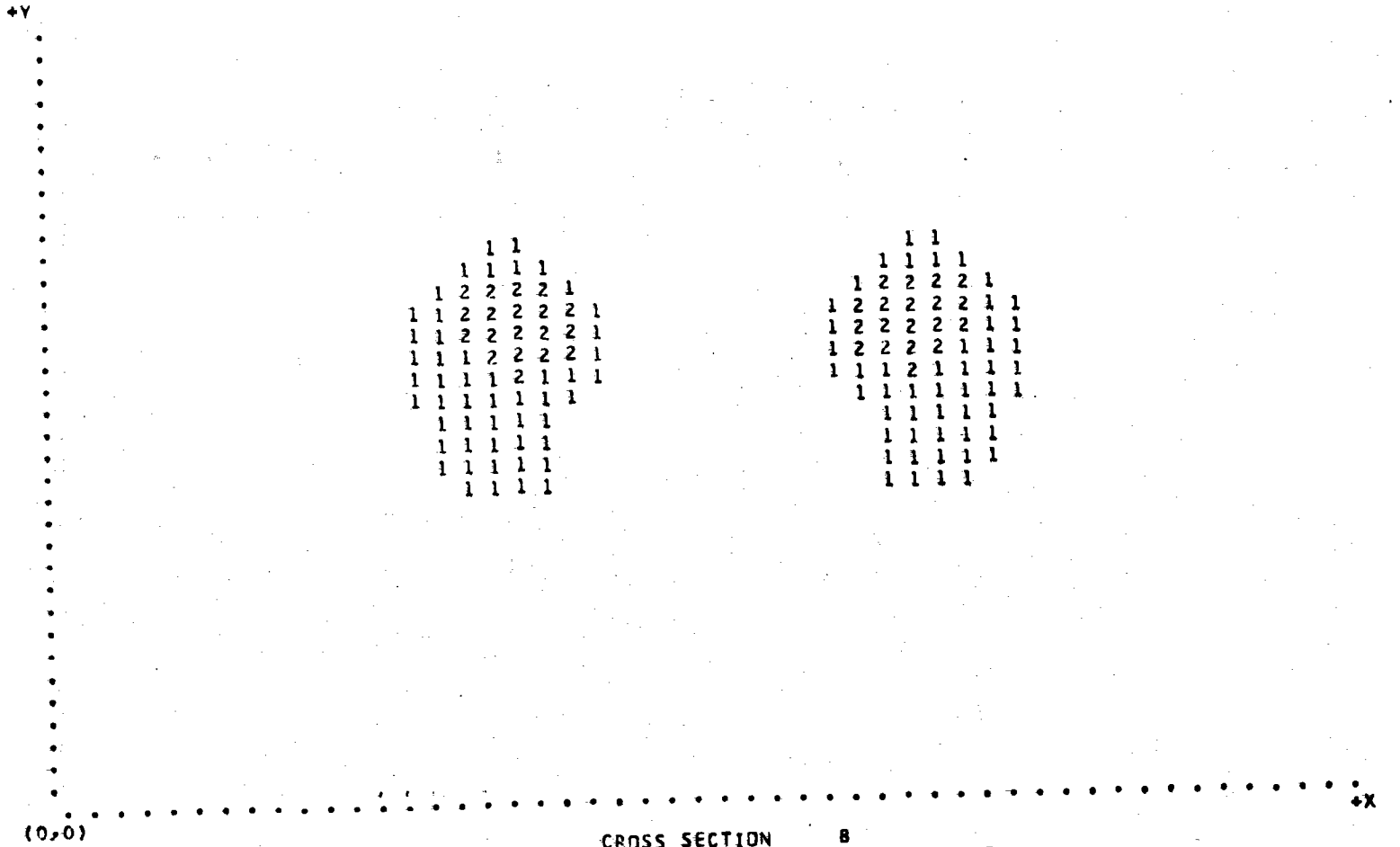
CROSS SECTION 5
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



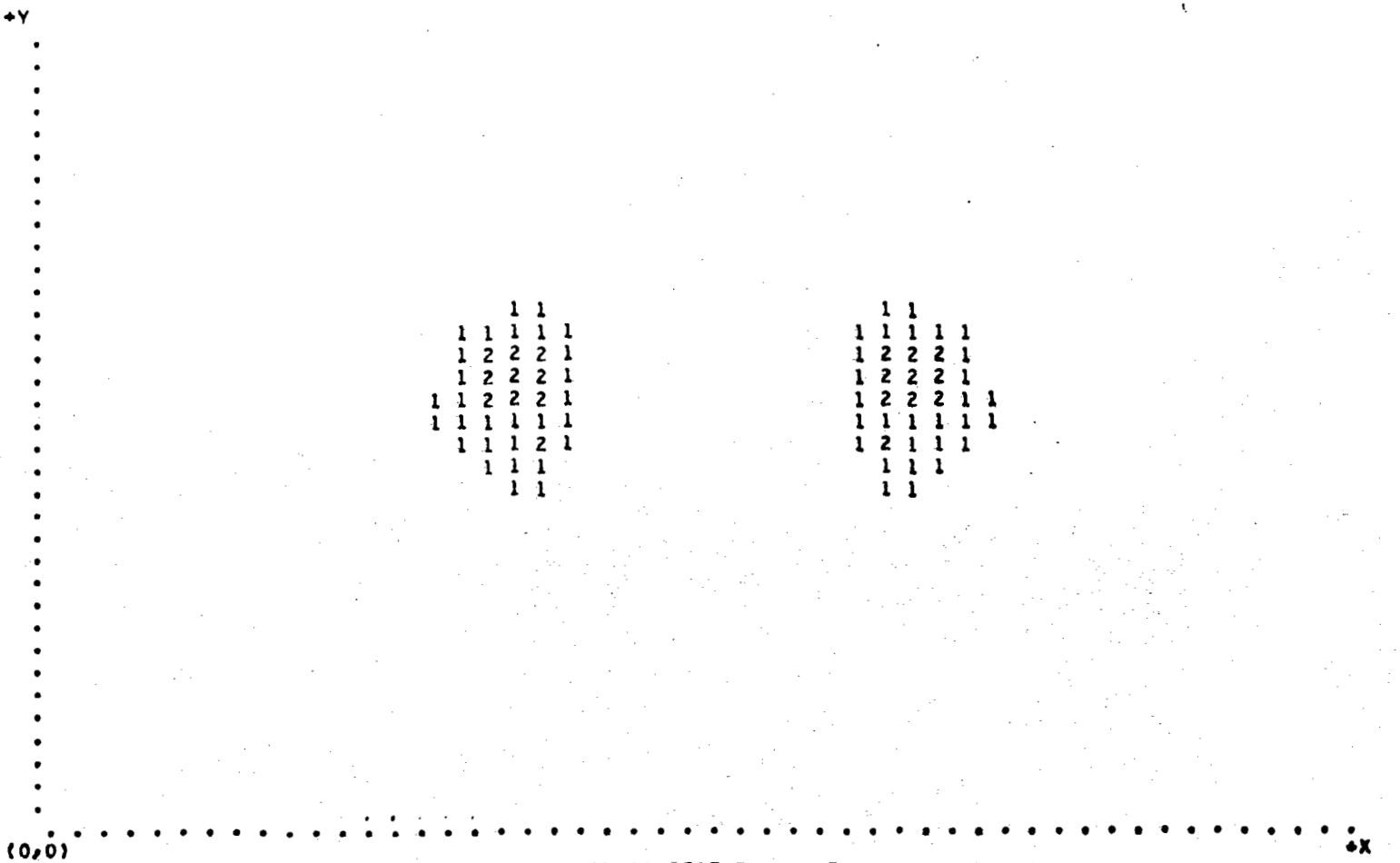
CROSS SECTION 6
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



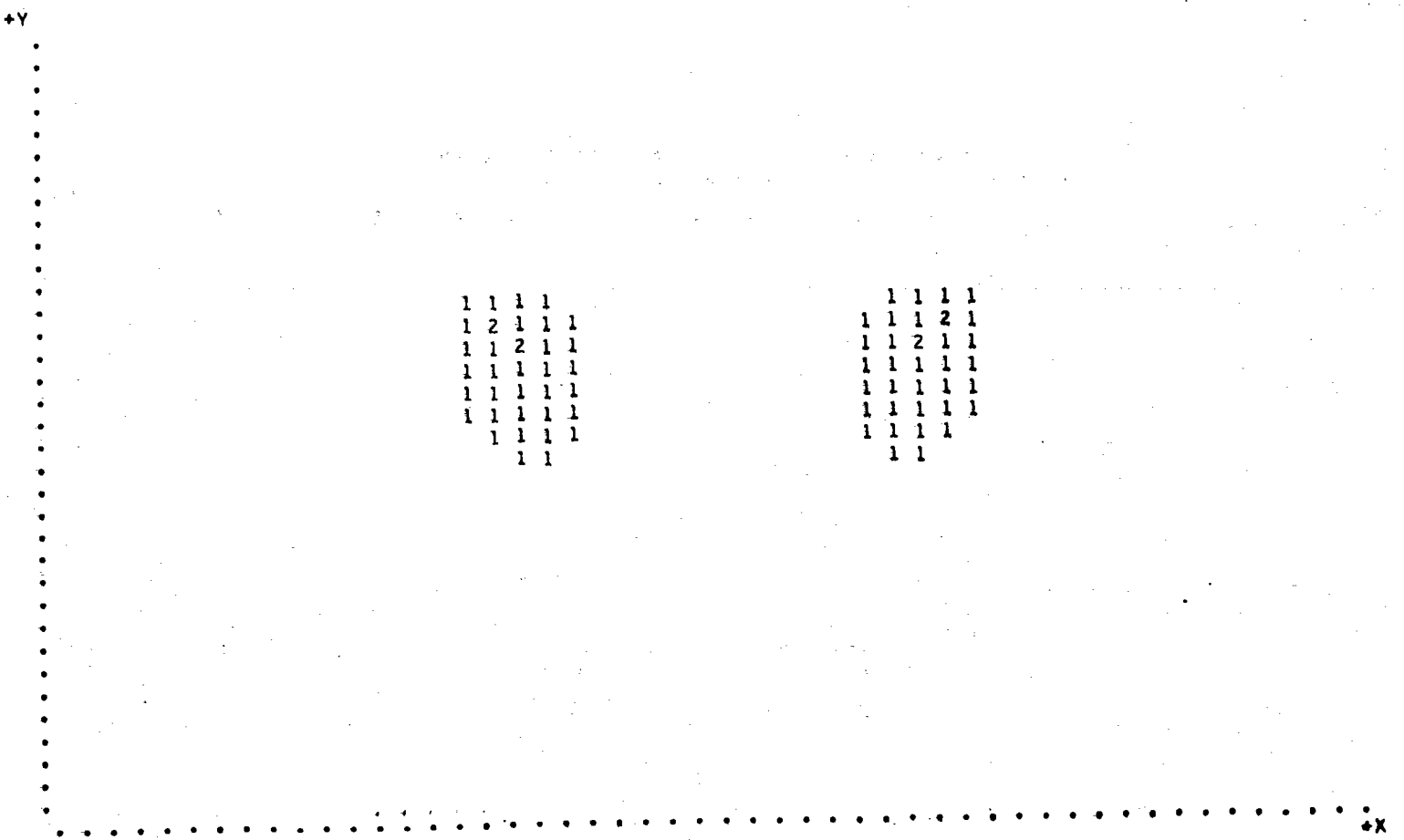
CROSS SECTION 7
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



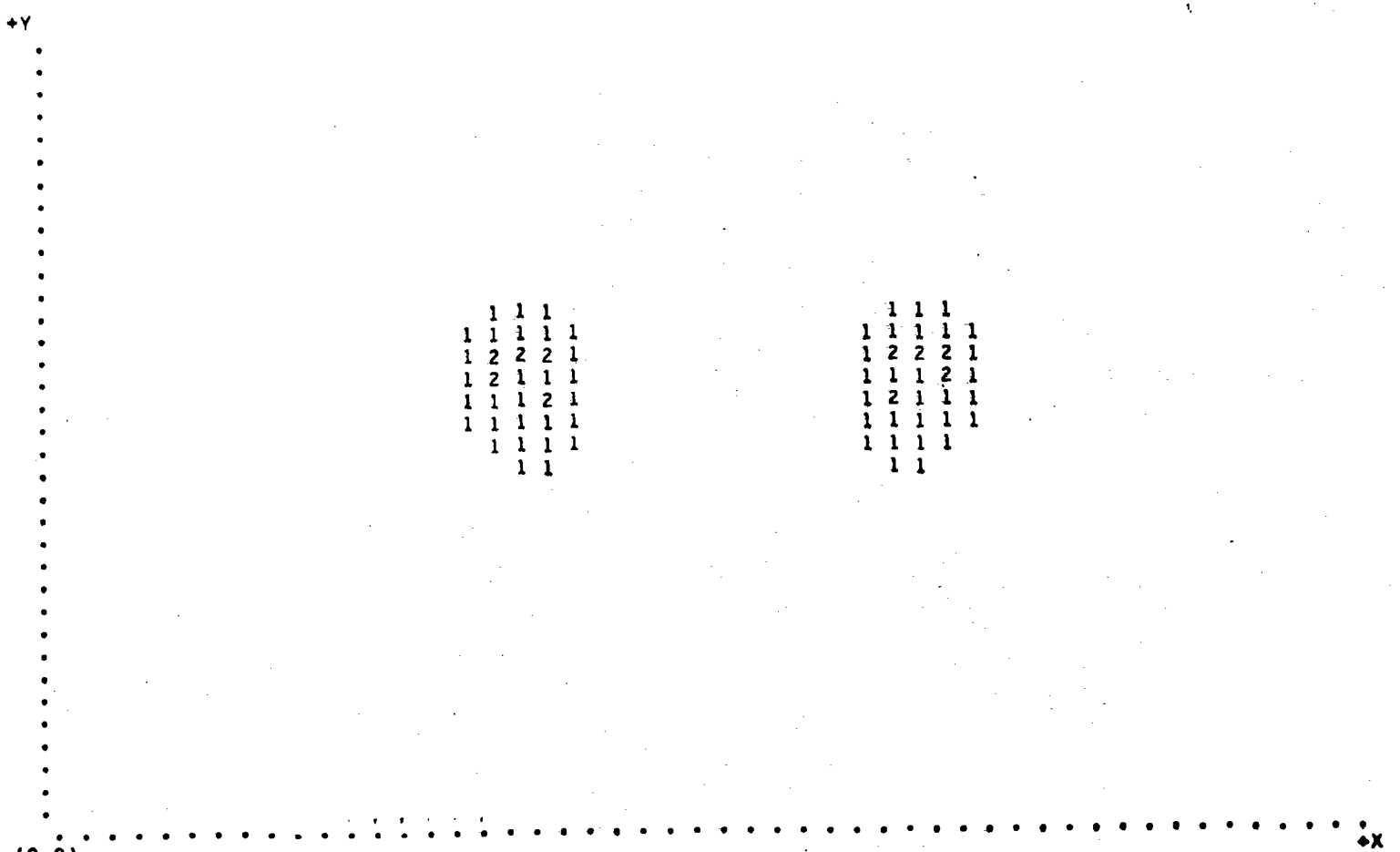
CROSS SECTION B
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



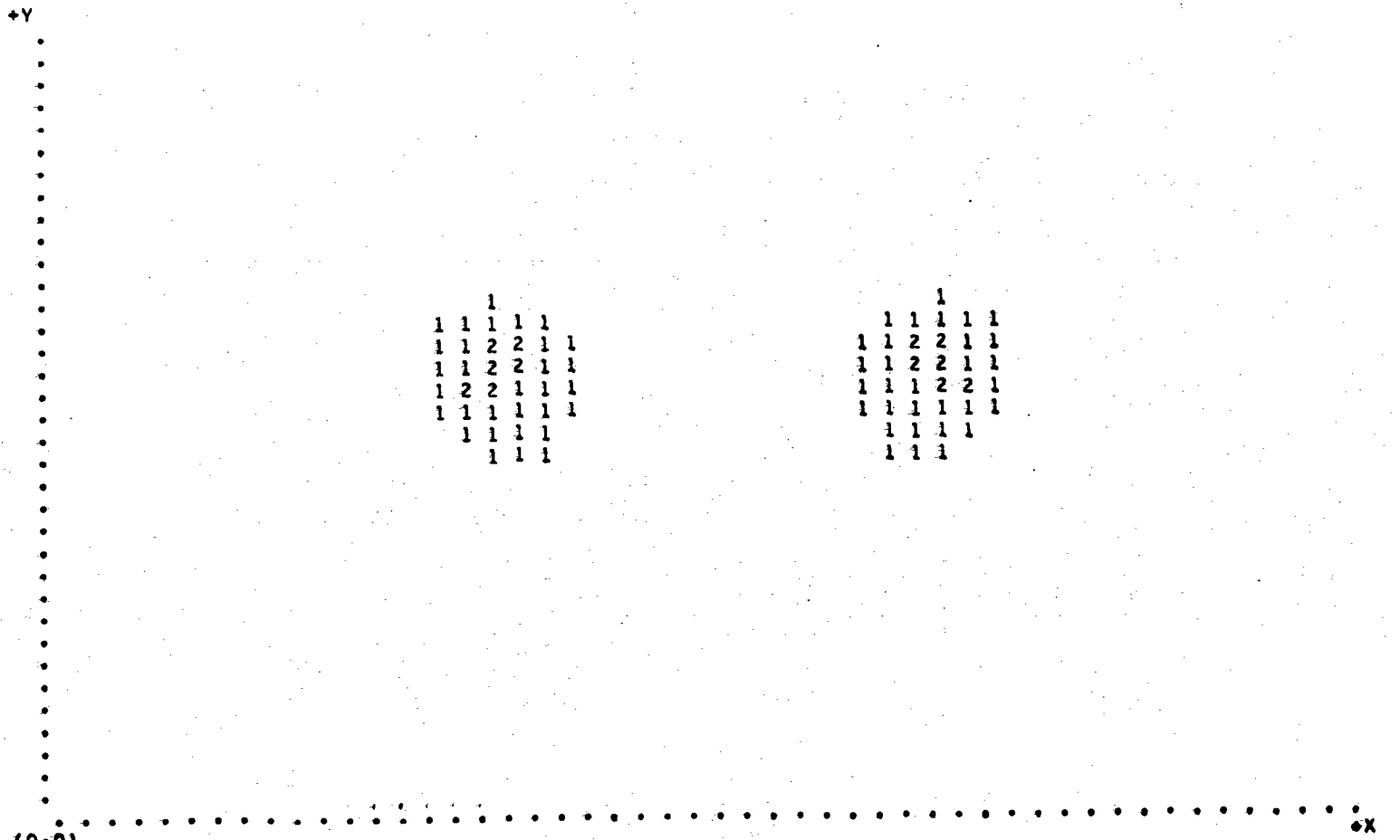
CROSS SECTION 9
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



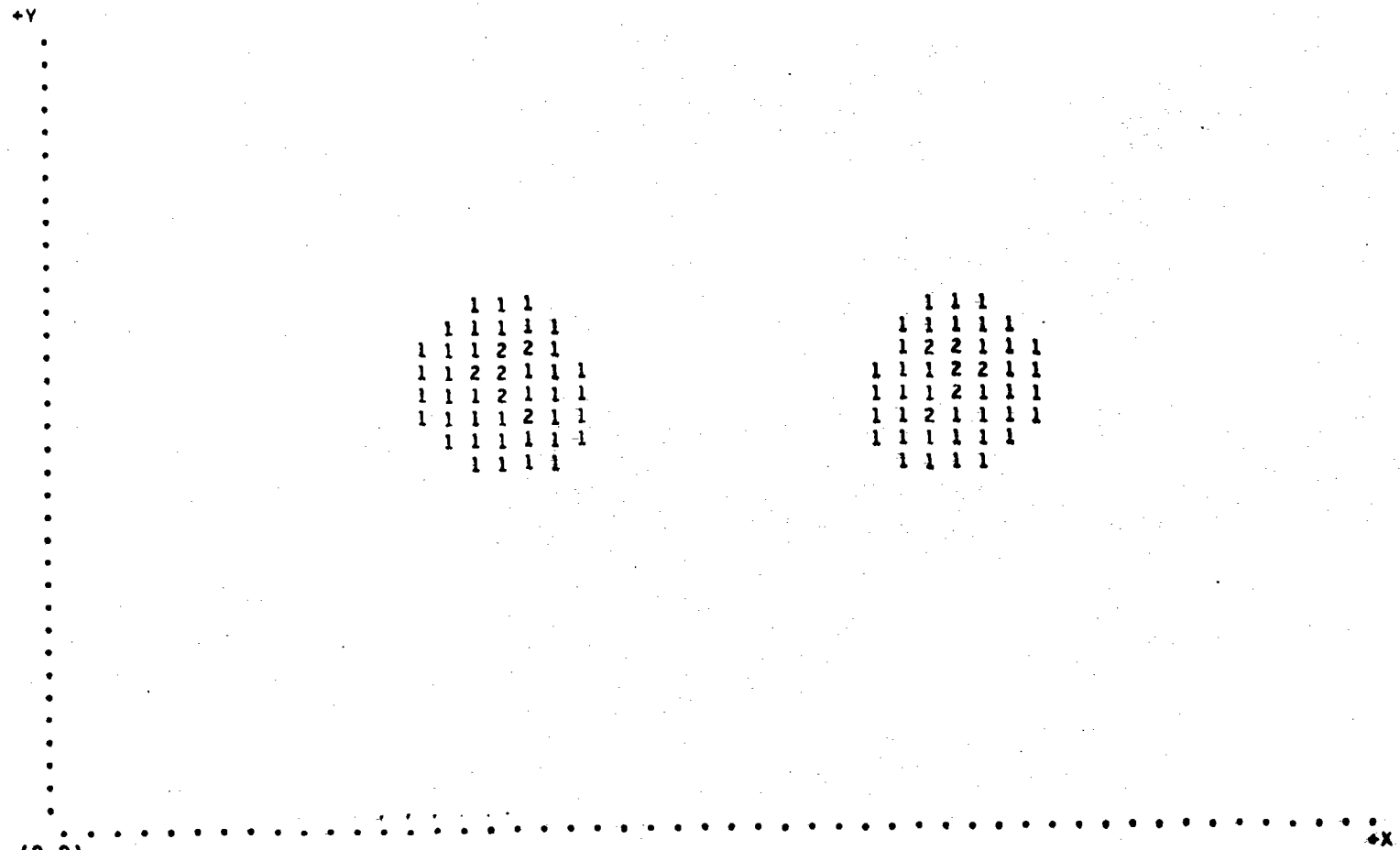
CROSS SECTION 10
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



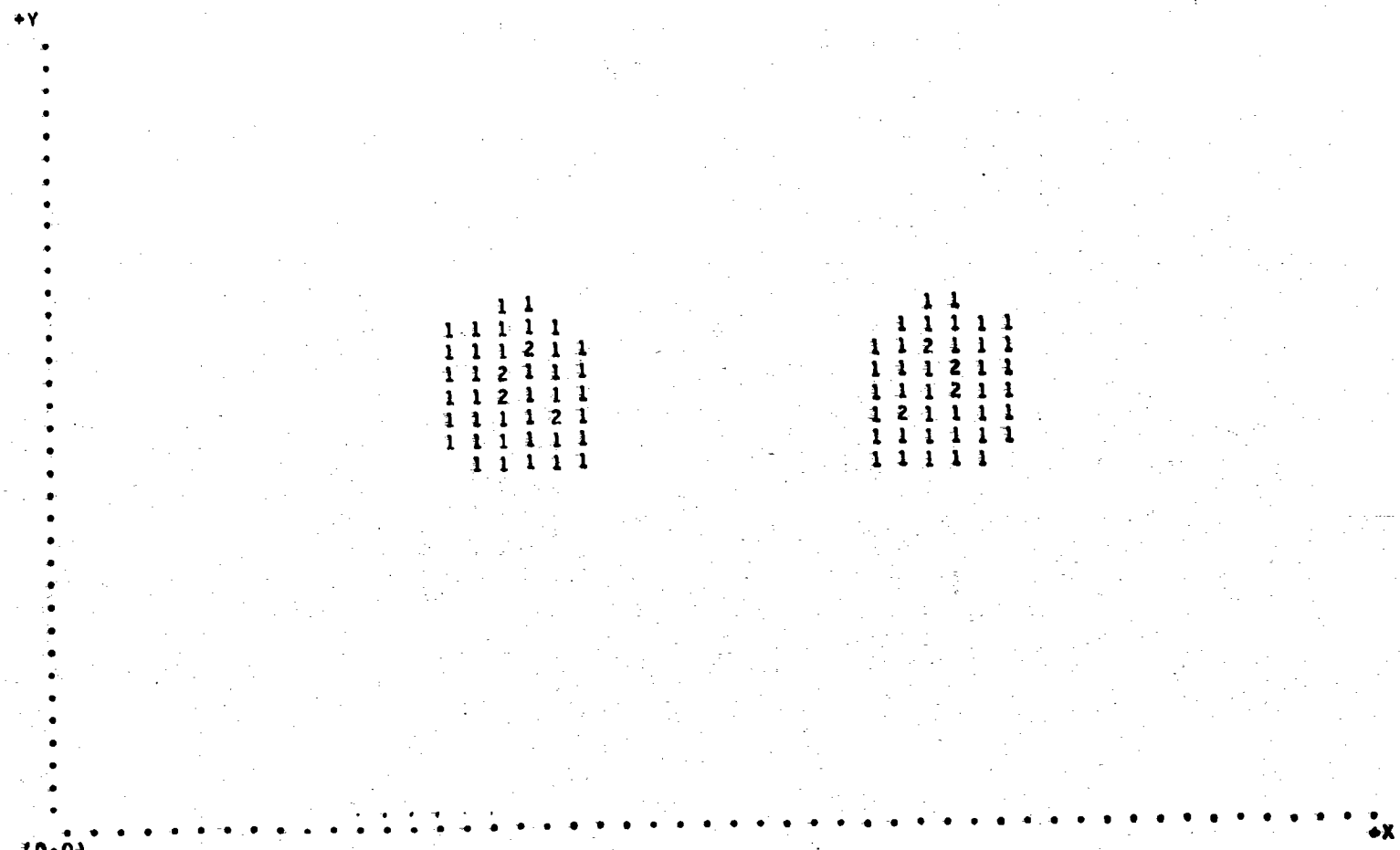
CROSS SECTION 11
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



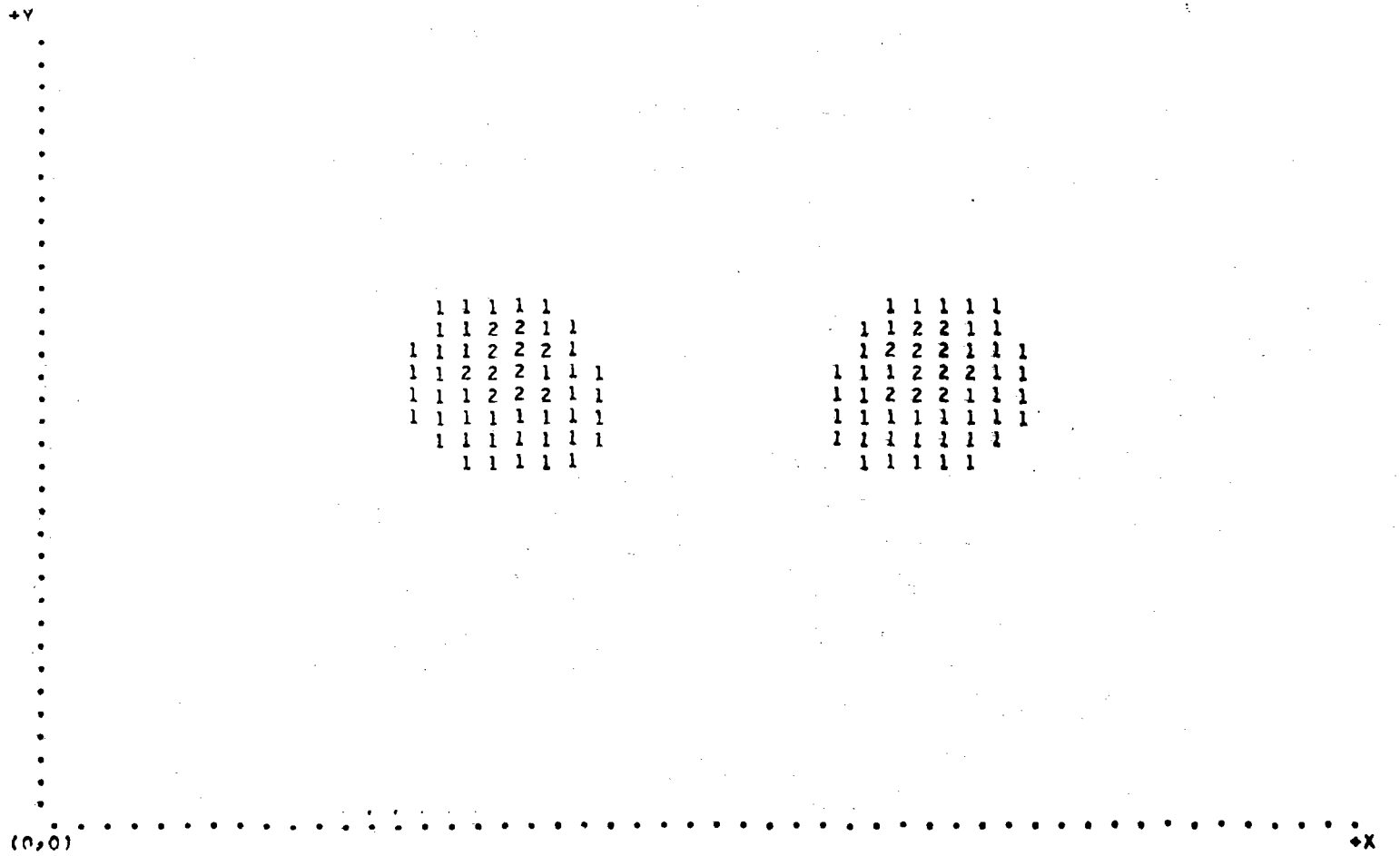
CROSS SECTION 12
AVERAGE DOCTURS EVALUATION 1 HOUR ASSESSMENT



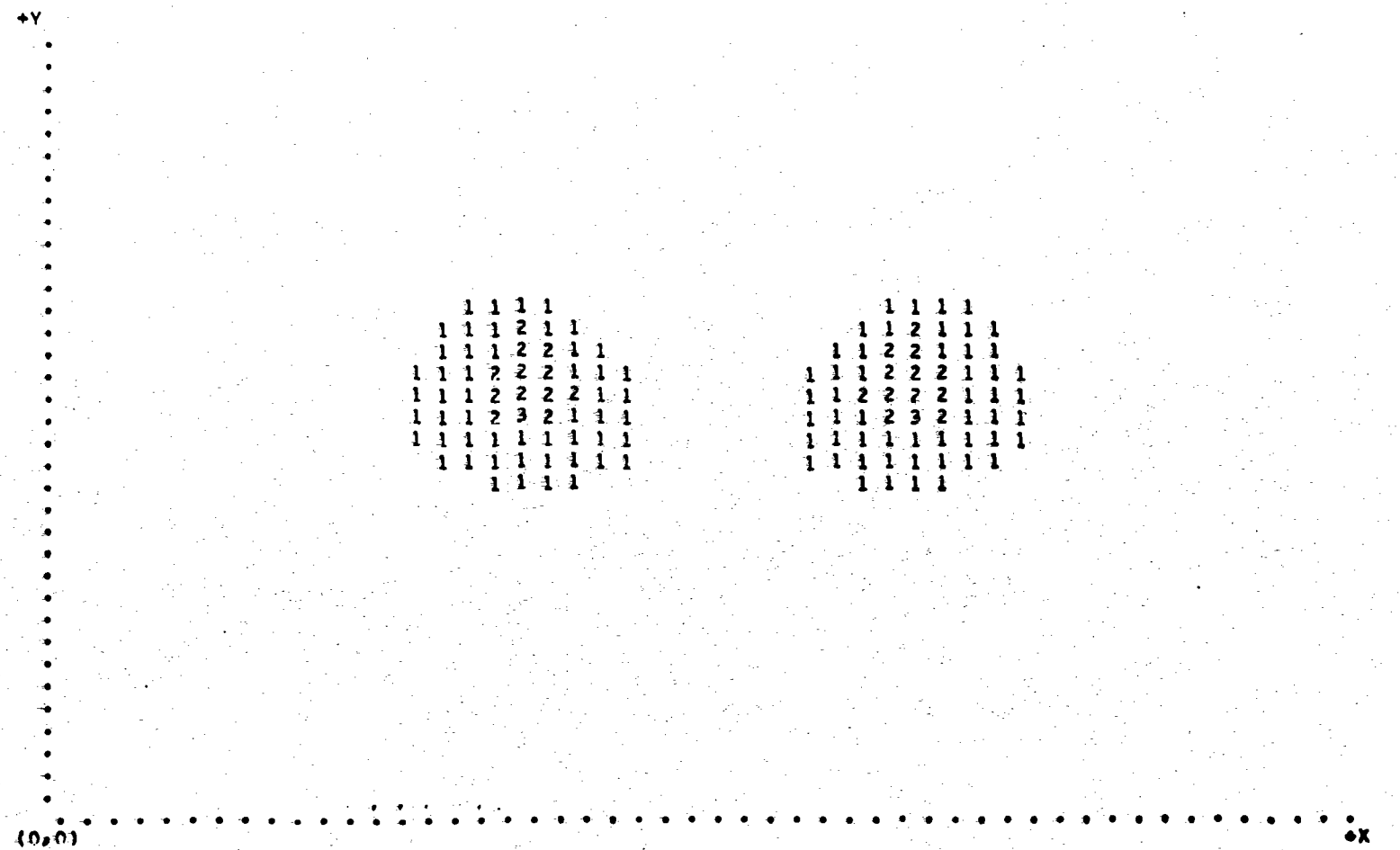
CROSS SECTION 13
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



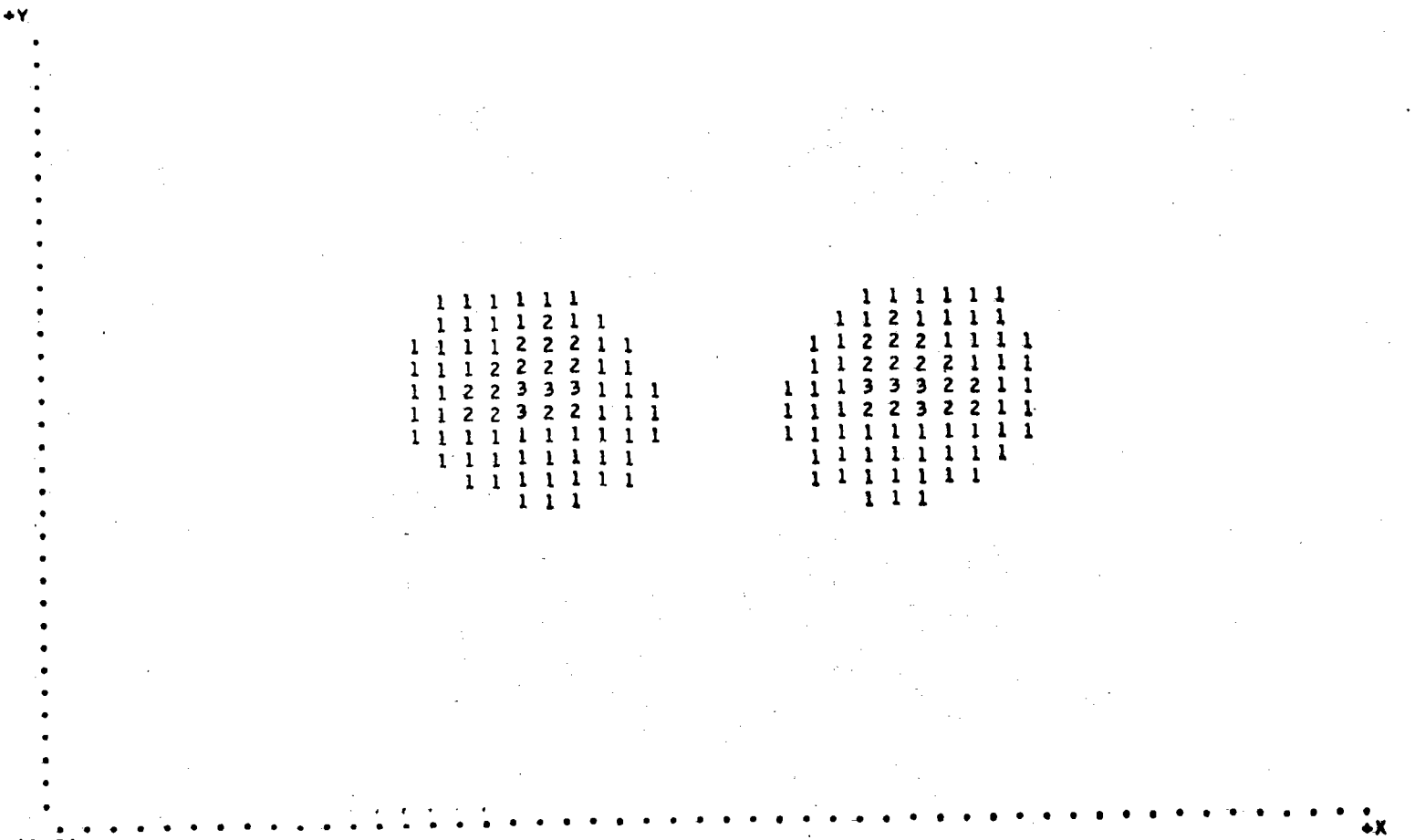
CROSS SECTION 14
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



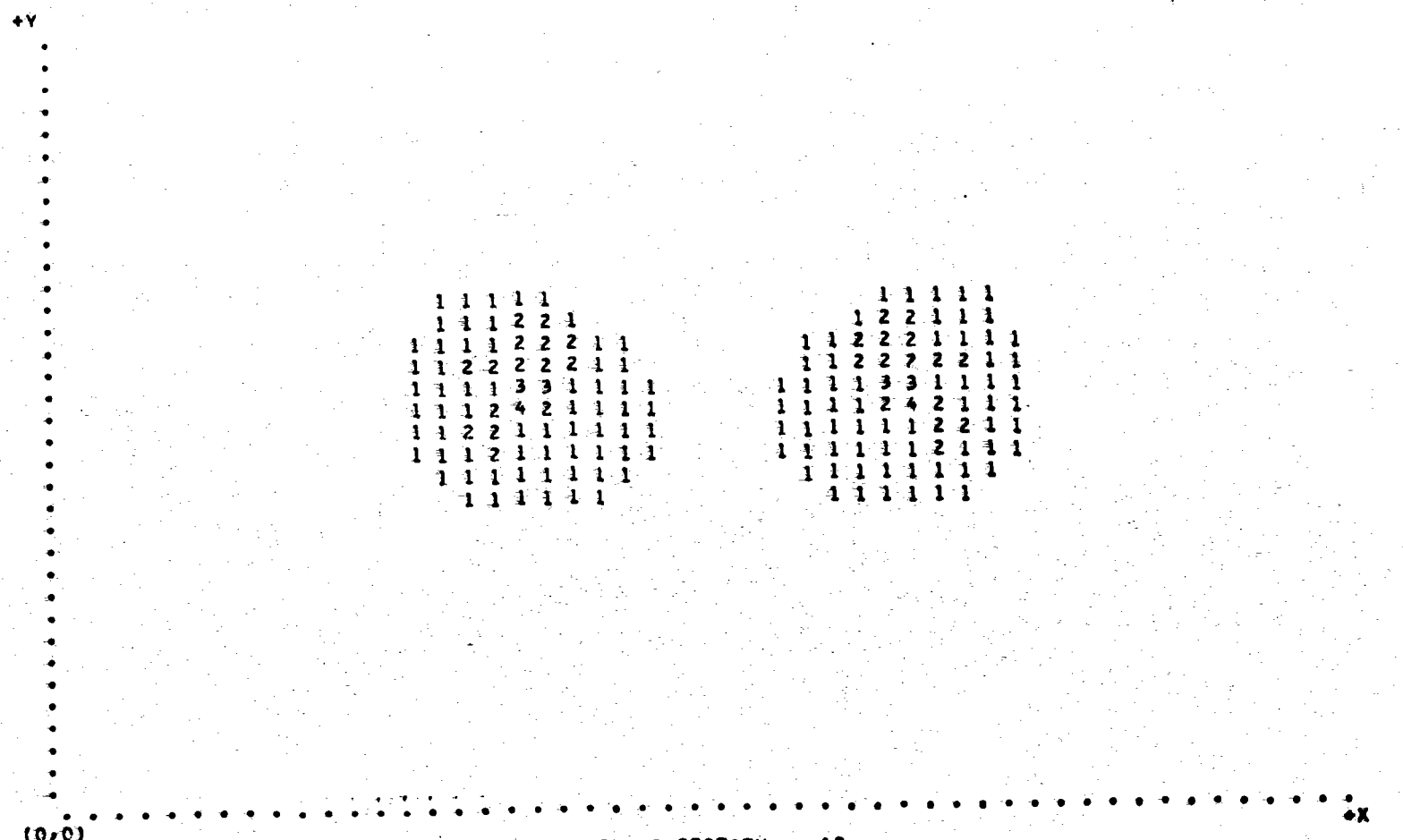
CROSS SECTION 15
AVERAGE DOCTURS EVALUATION 1 HOUR ASSESSMENT



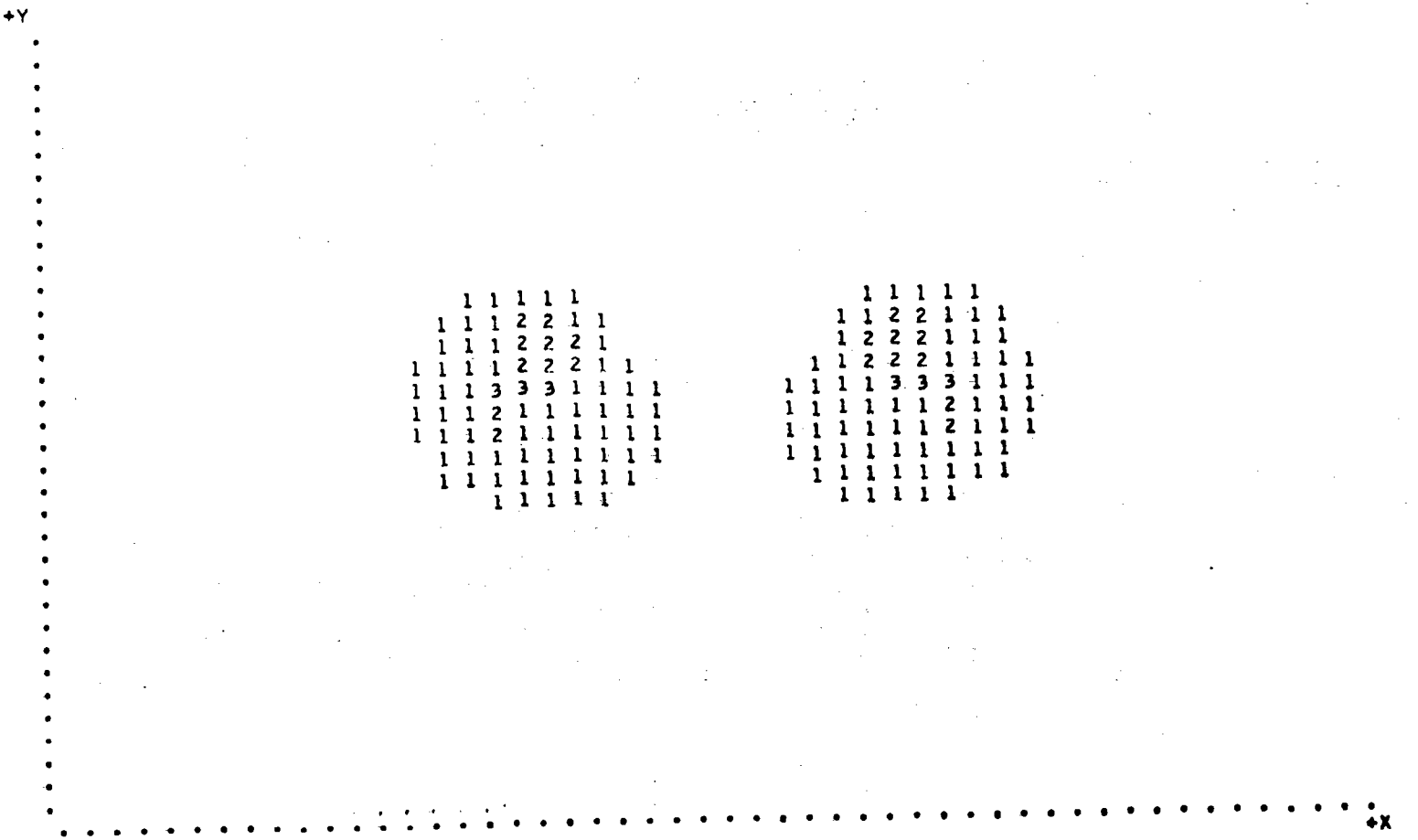
CROSS SECTION 16
AVERAGE DOCTURS EVALUATION 1 HOUR ASSESSMENT



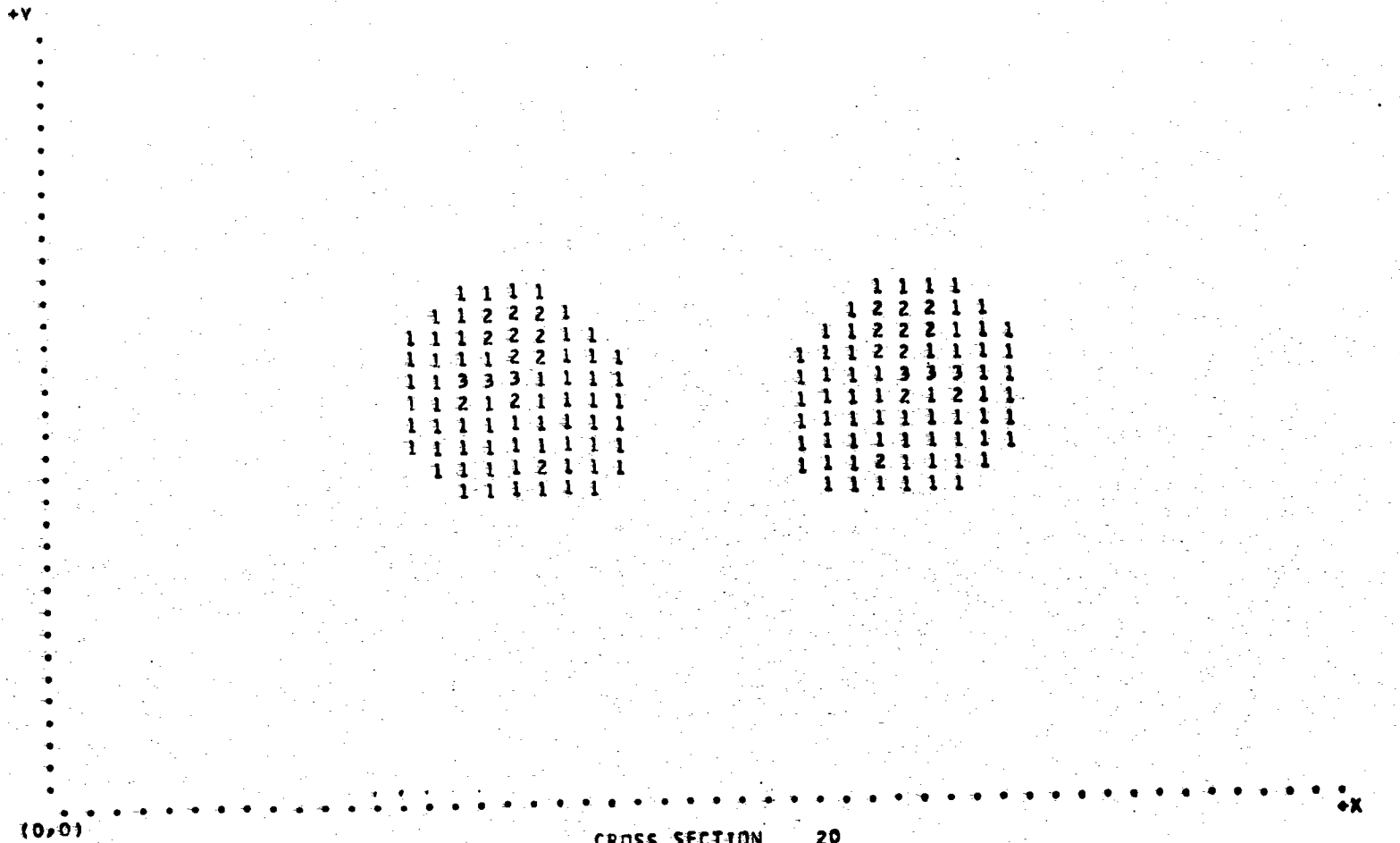
CROSS SECTION 17
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



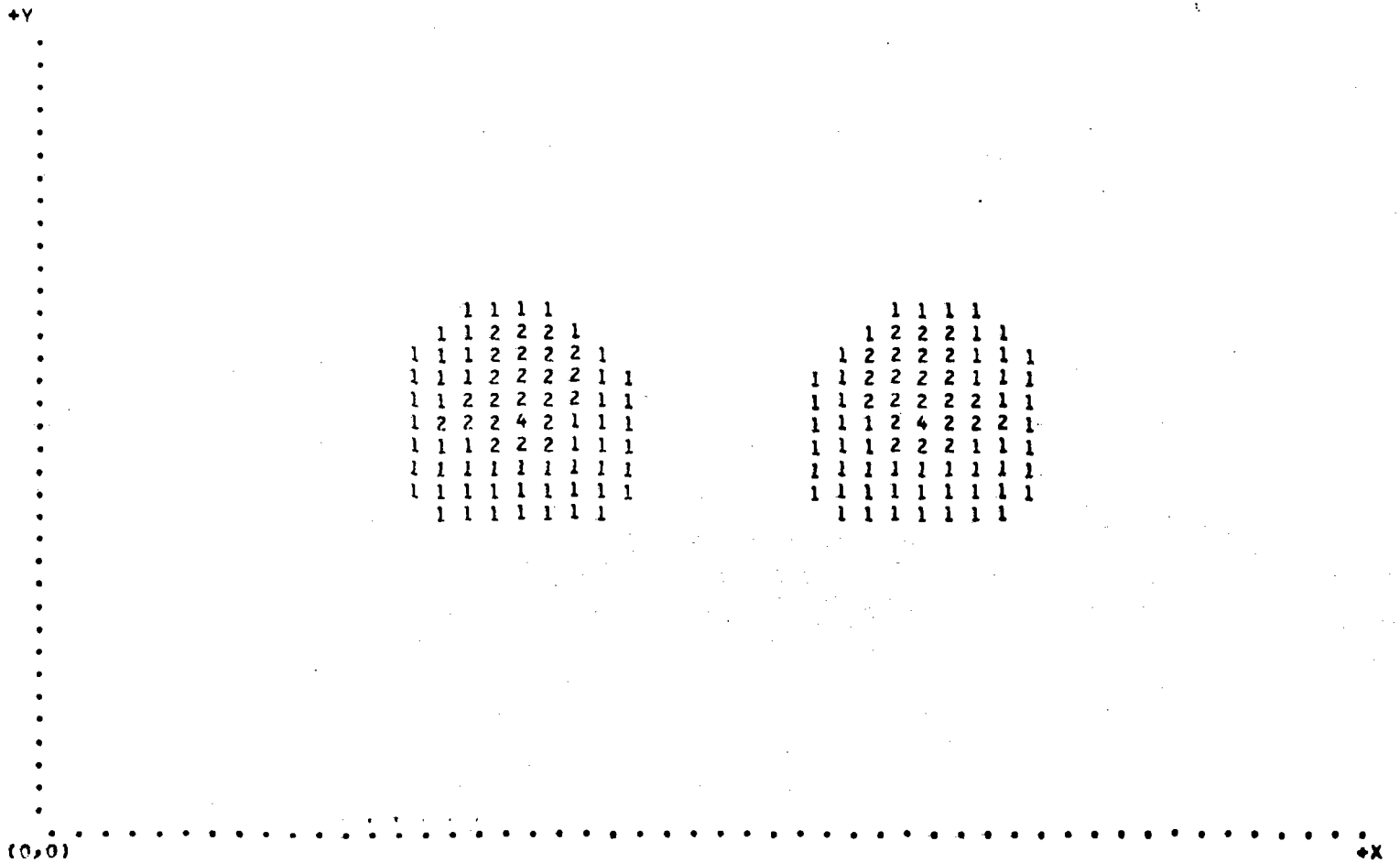
CROSS SECTION 18
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



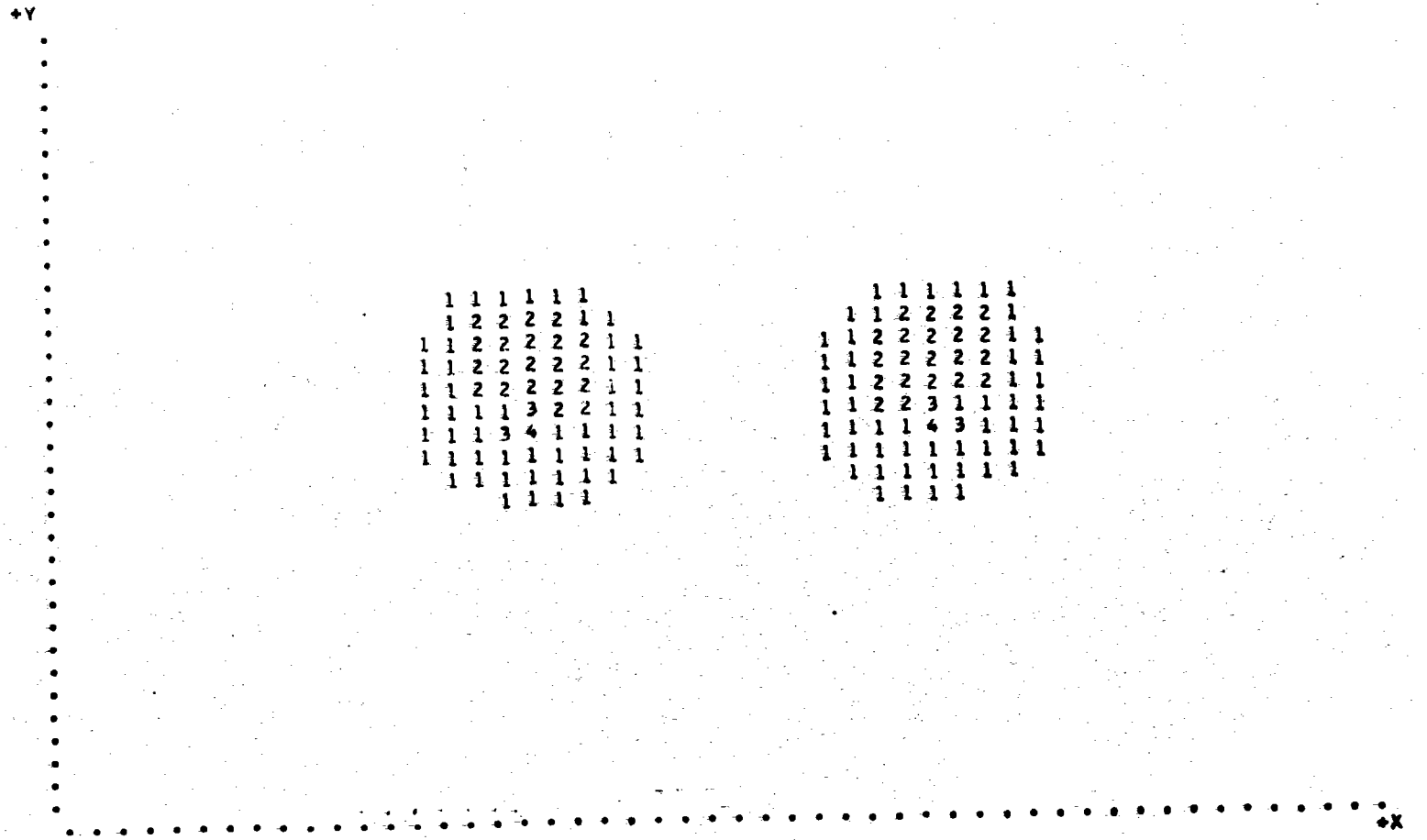
CROSS SECTION 19
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



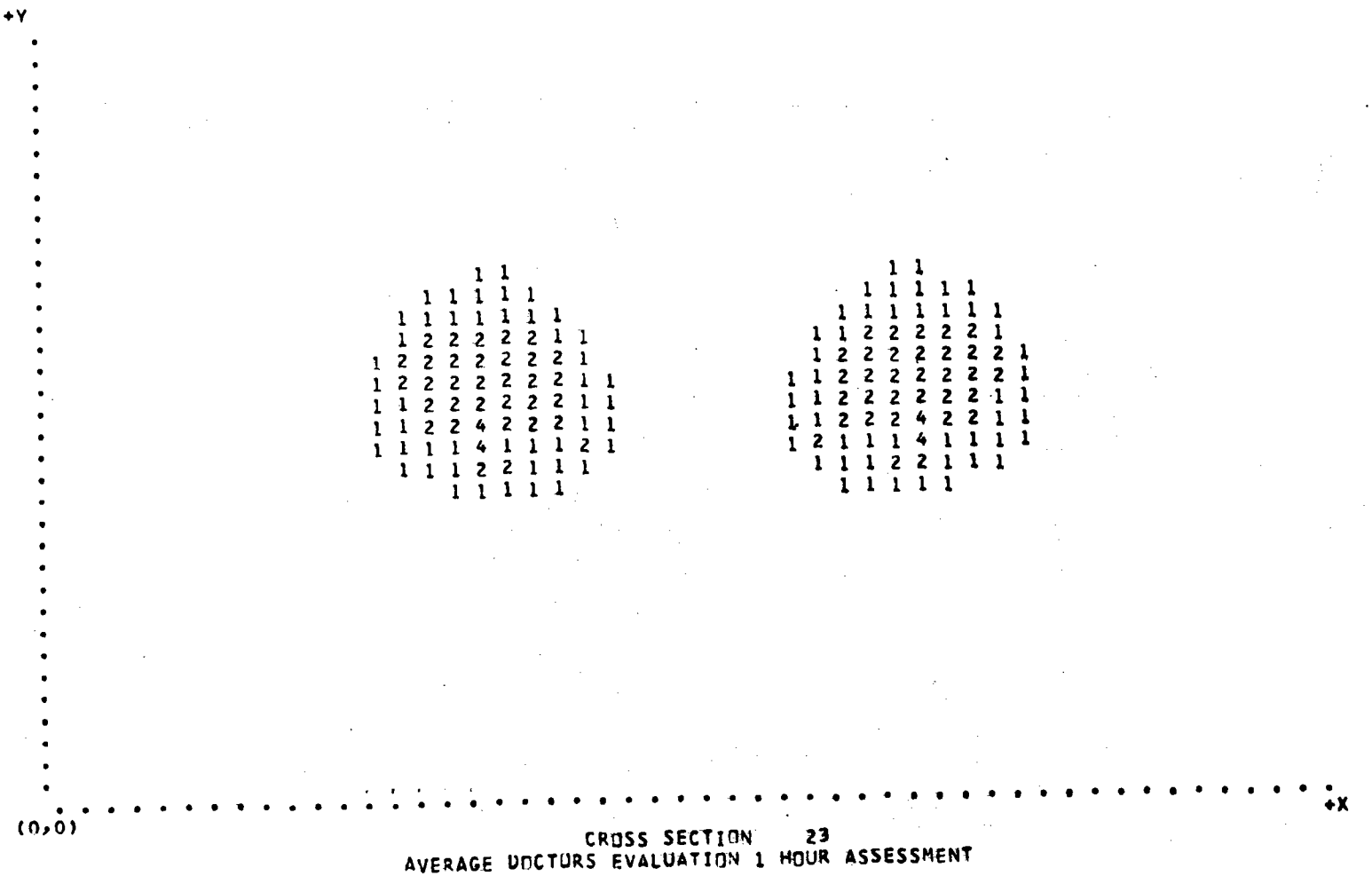
CROSS SECTION 20
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



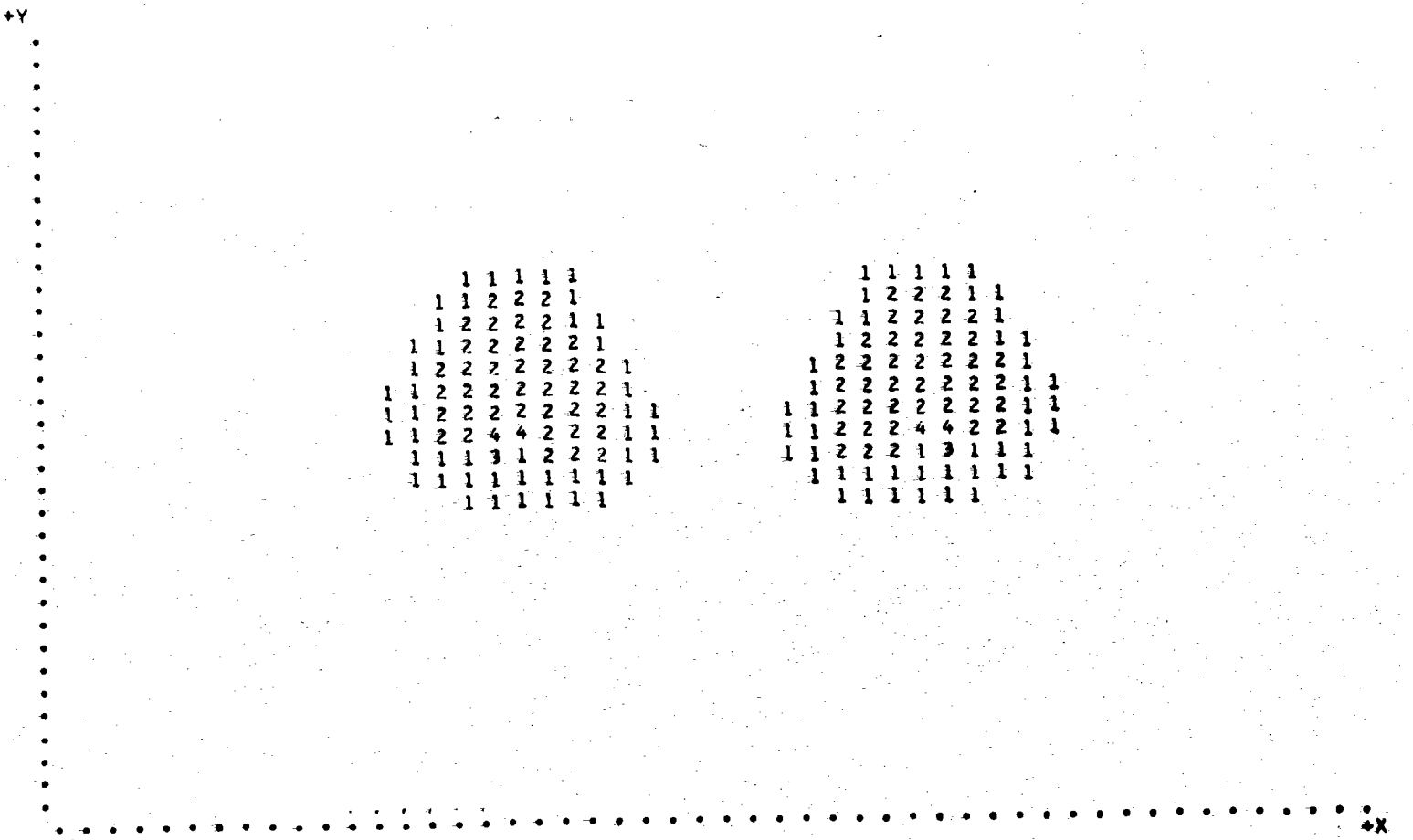
CROSS SECTION 21
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



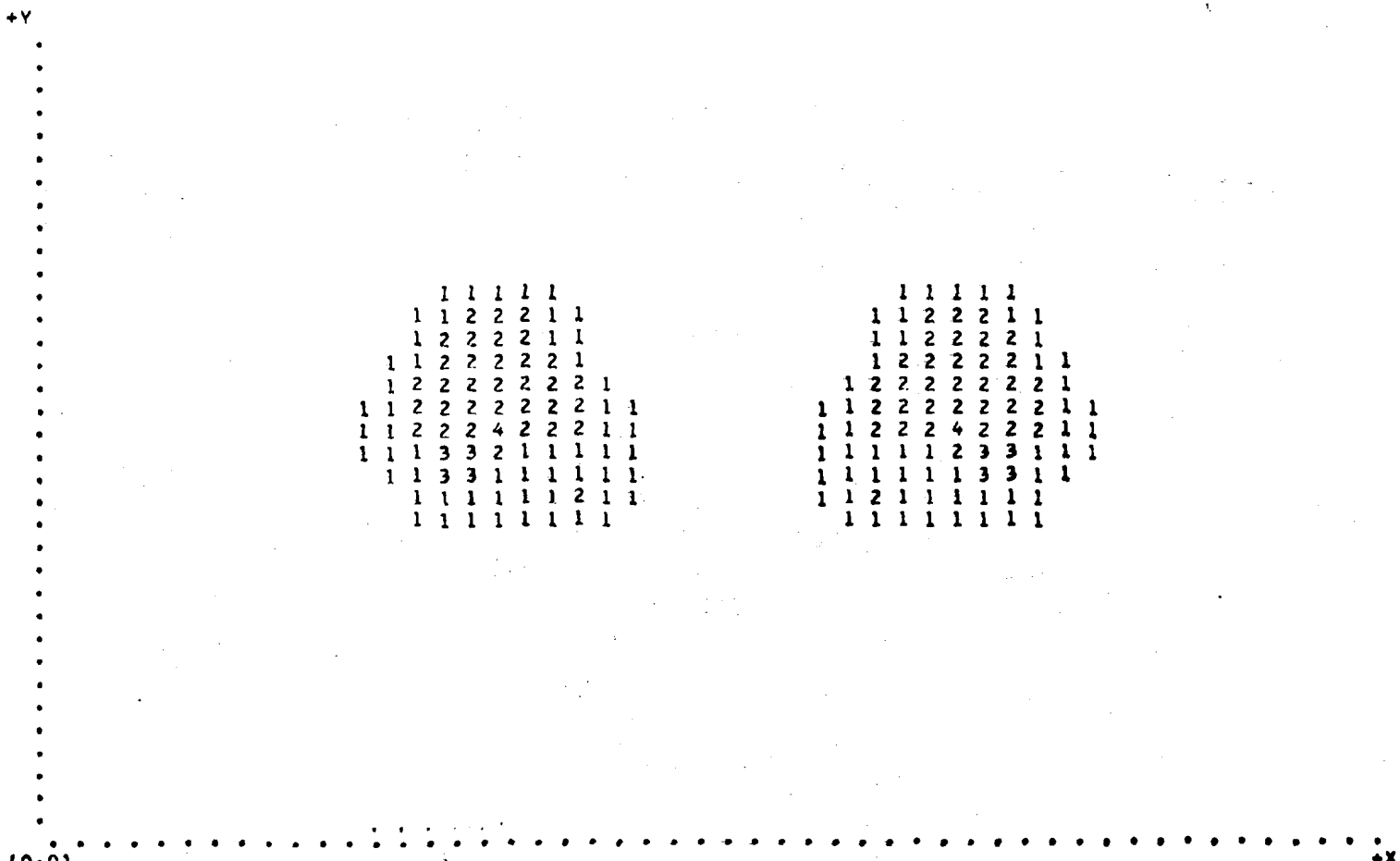
CROSS SECTION 22
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



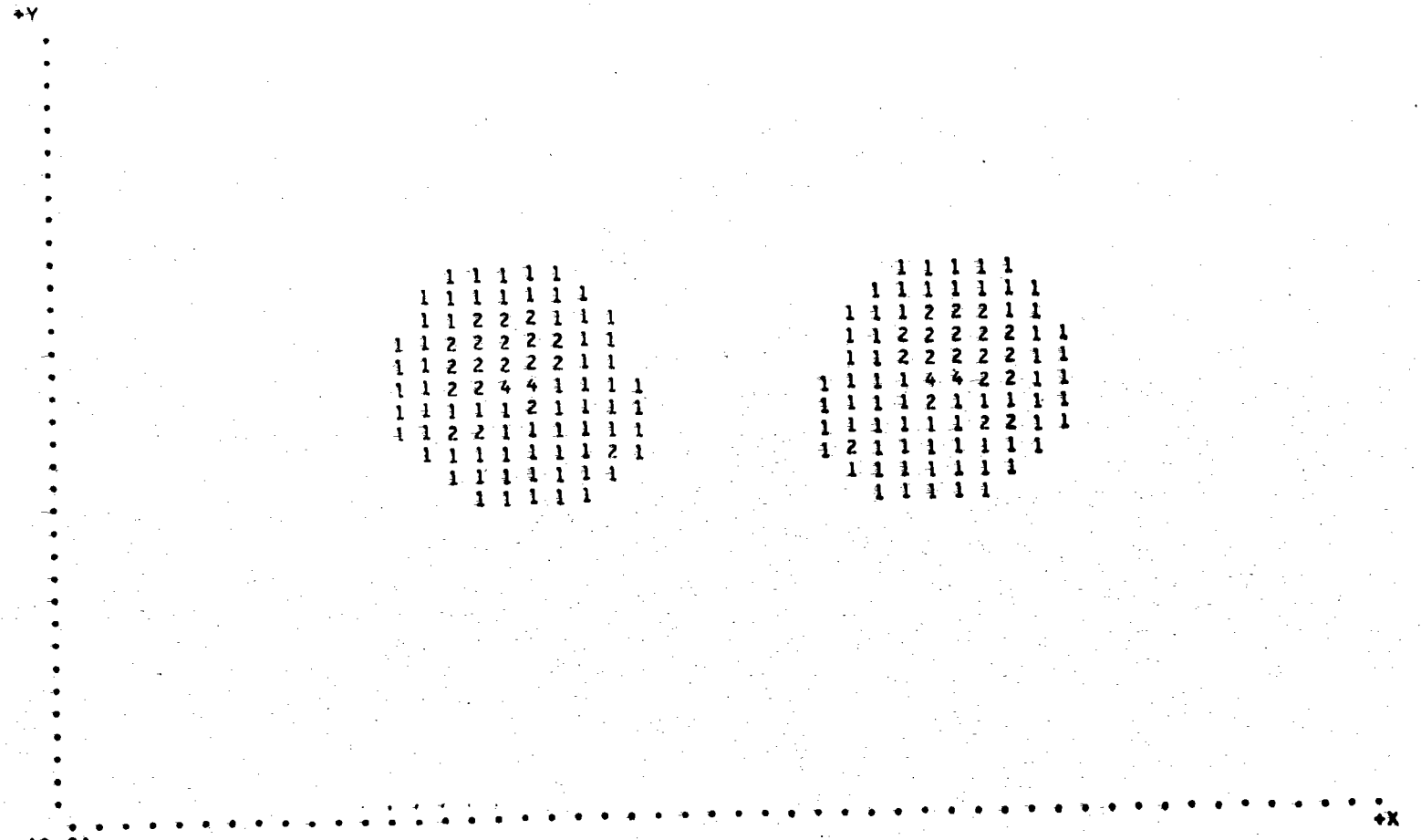
CROSS SECTION 23
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



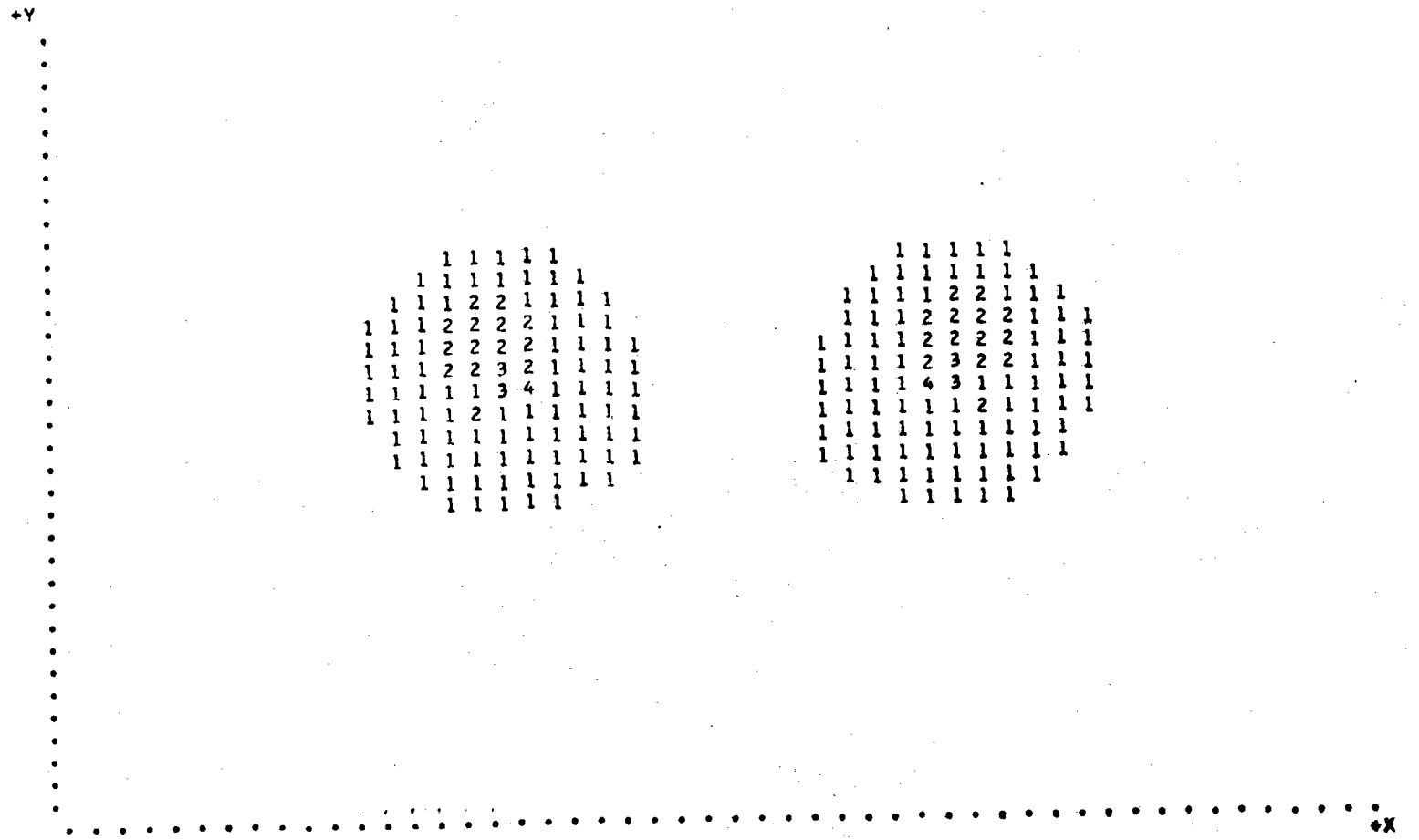
CROSS SECTION 24
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



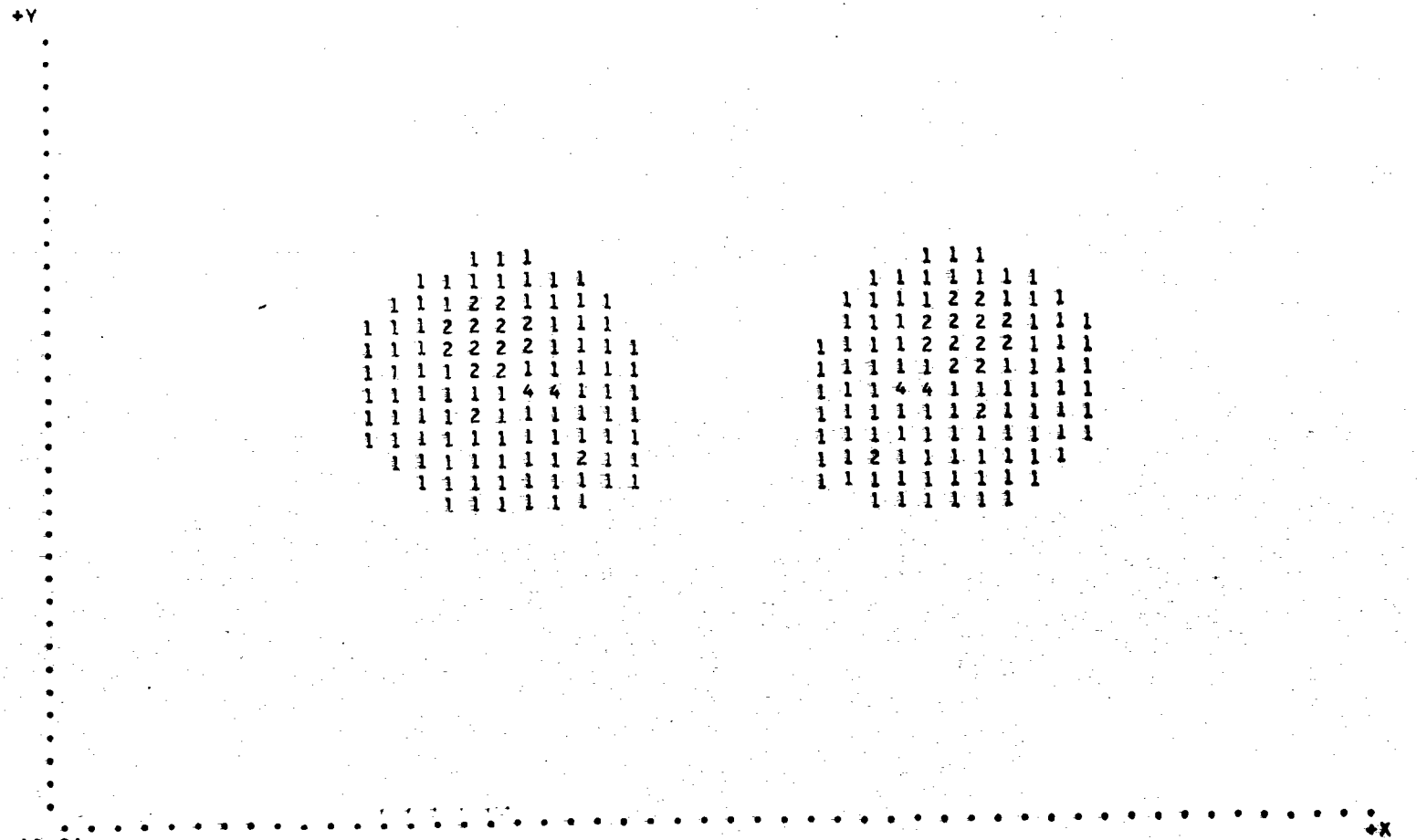
CROSS SECTION 25
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



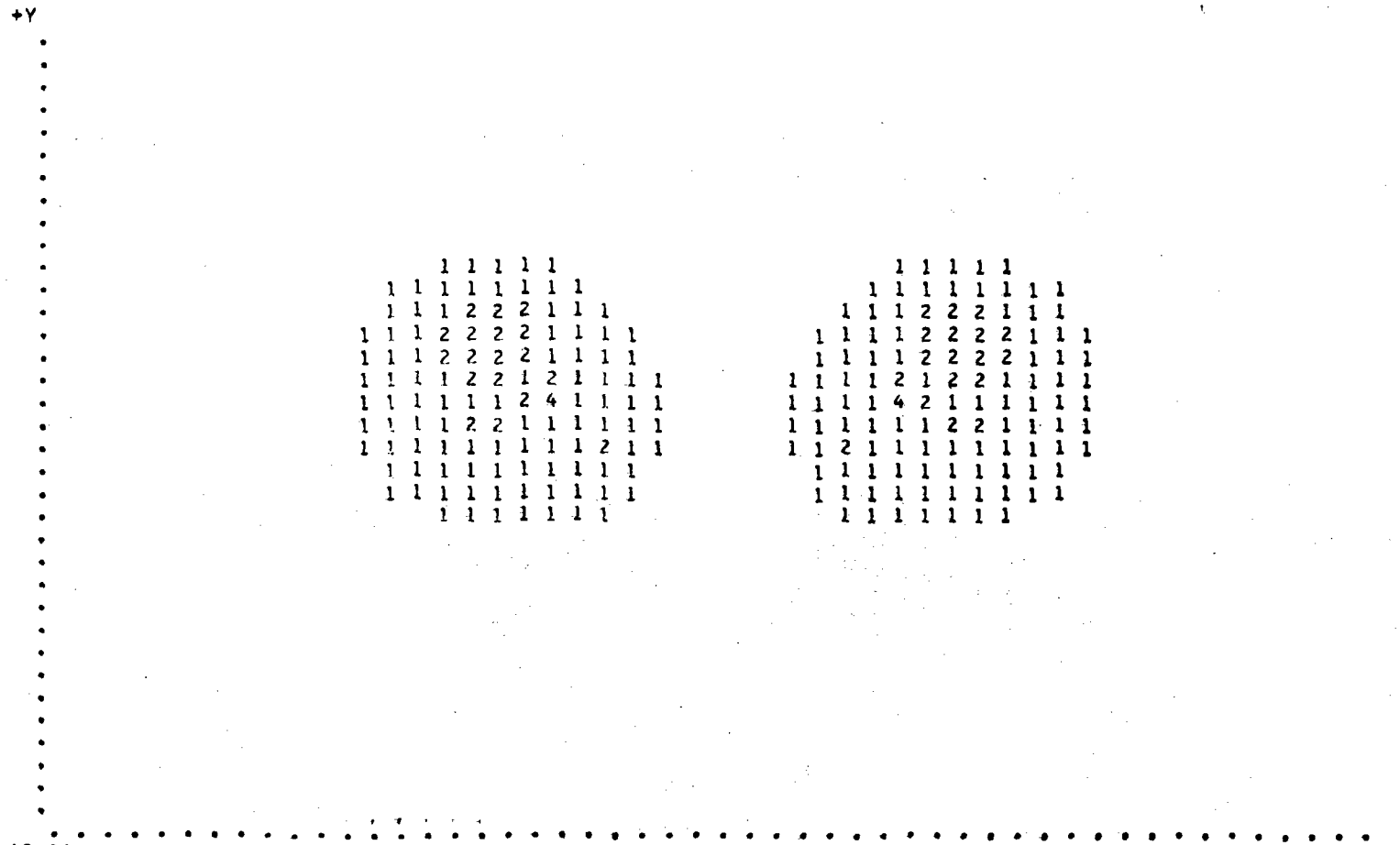
CROSS SECTION 26
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



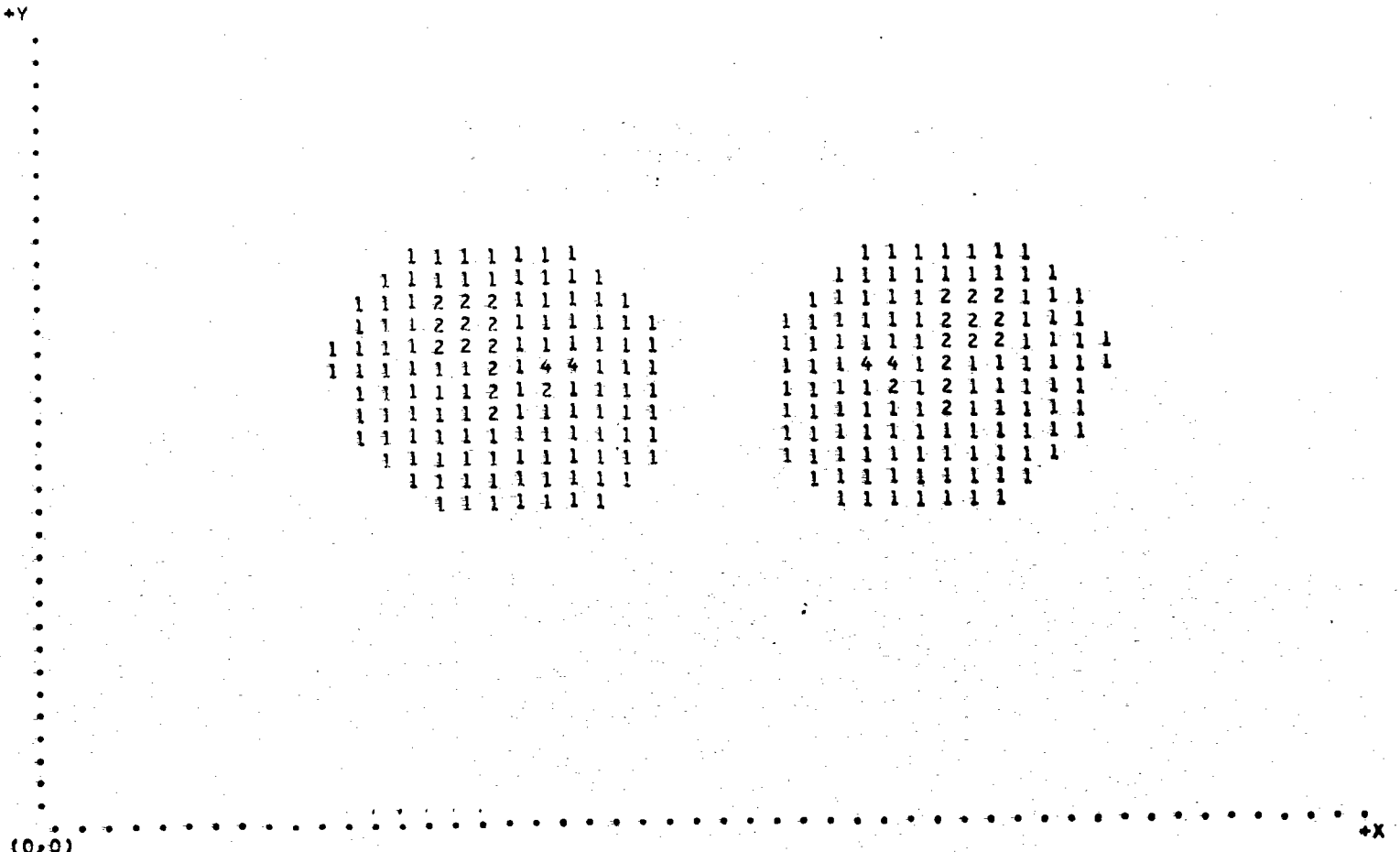
CROSS SECTION 27
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



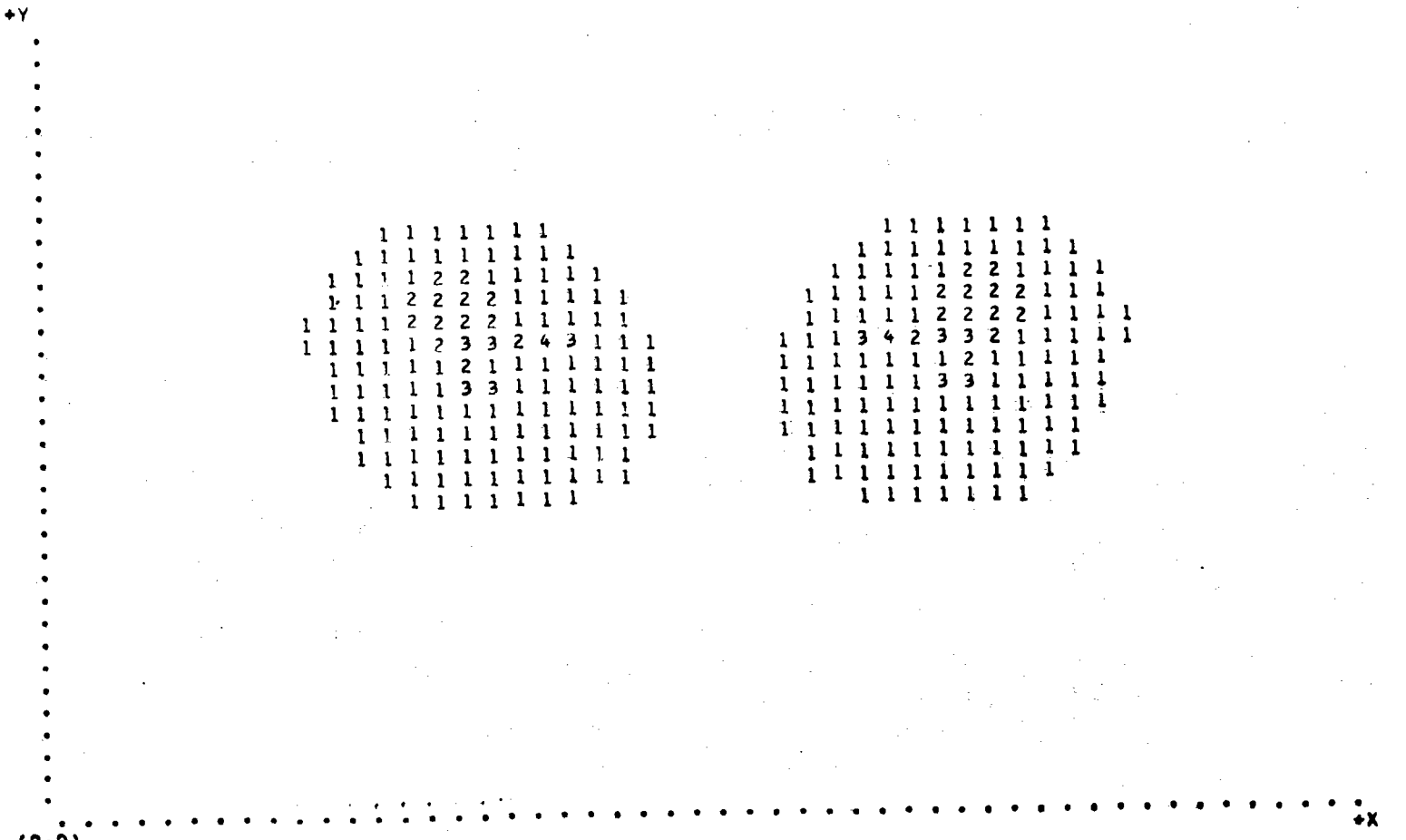
CROSS SECTION 28
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



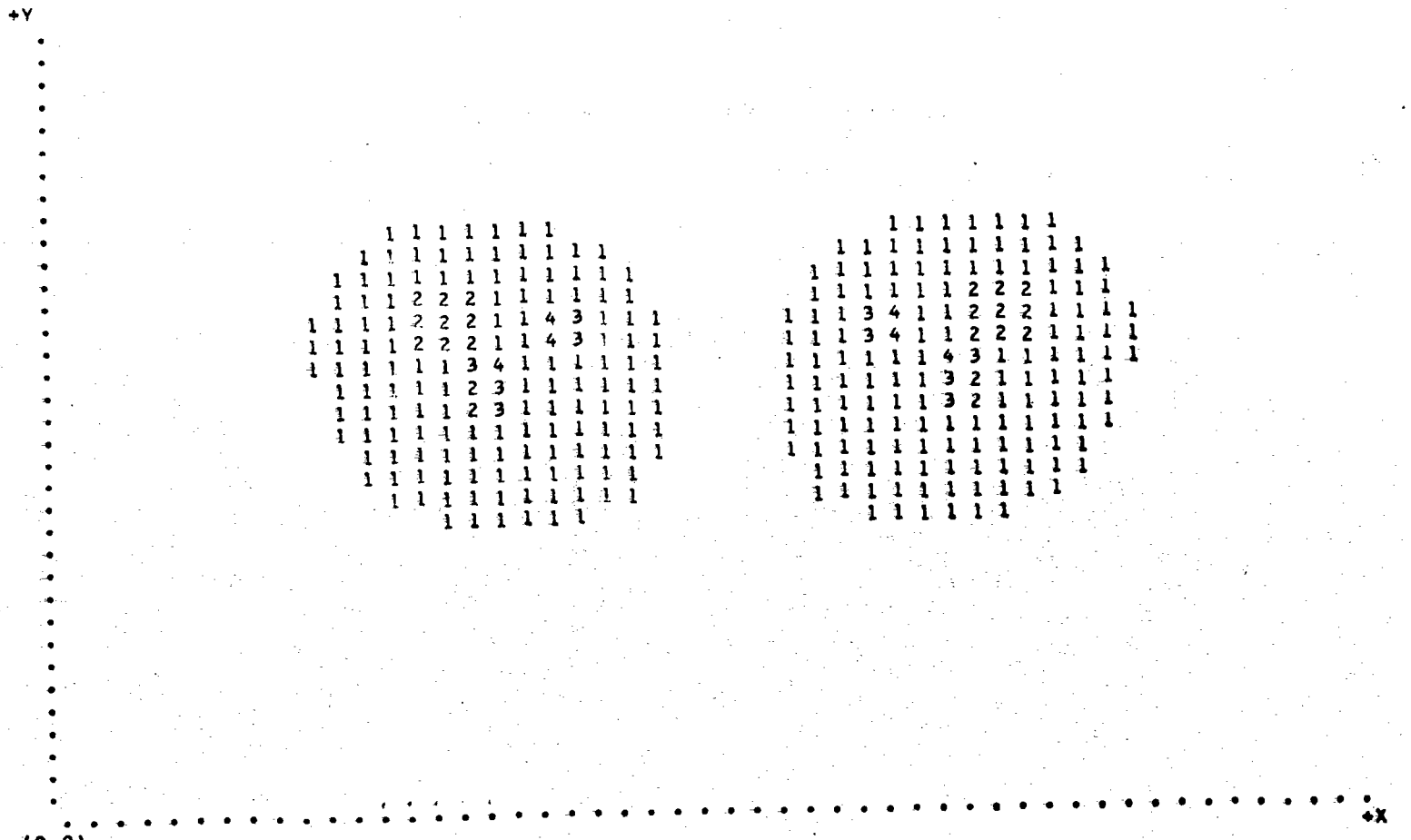
CROSS SECTION 29
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



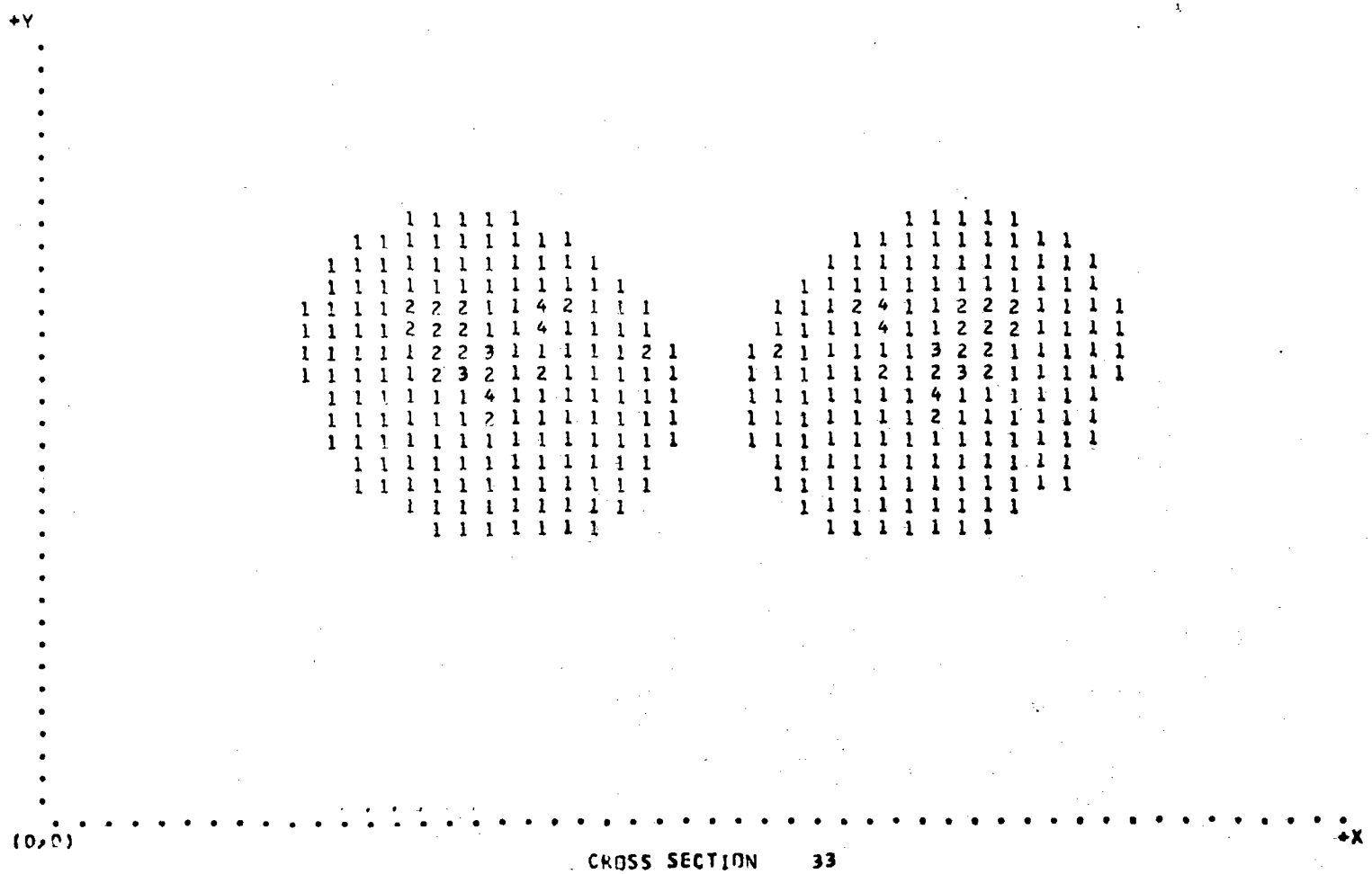
CROSS SECTION 30
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



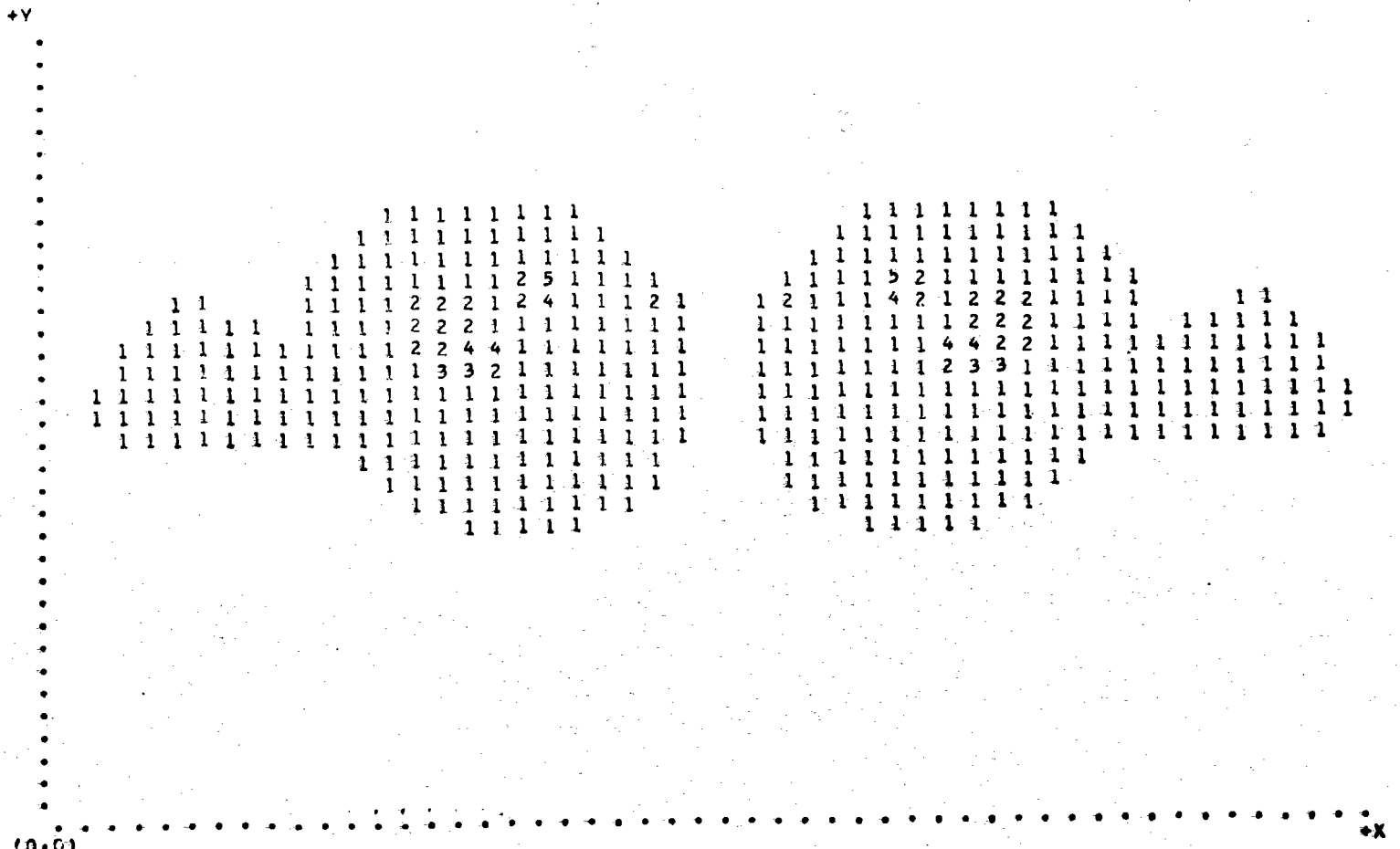
CROSS SECTION 31
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



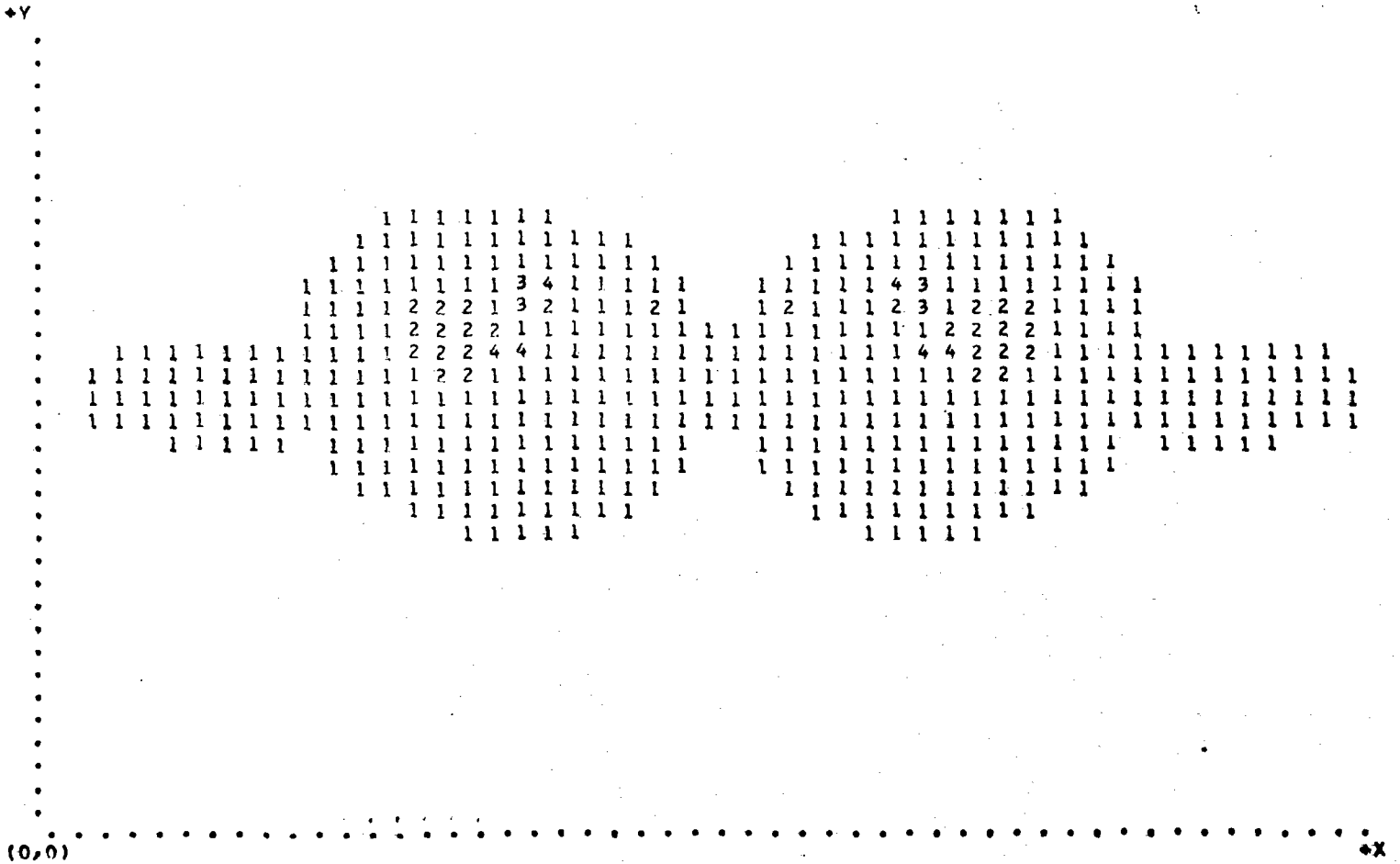
CROSS SECTION 32
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



CROSS SECTION 33
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT

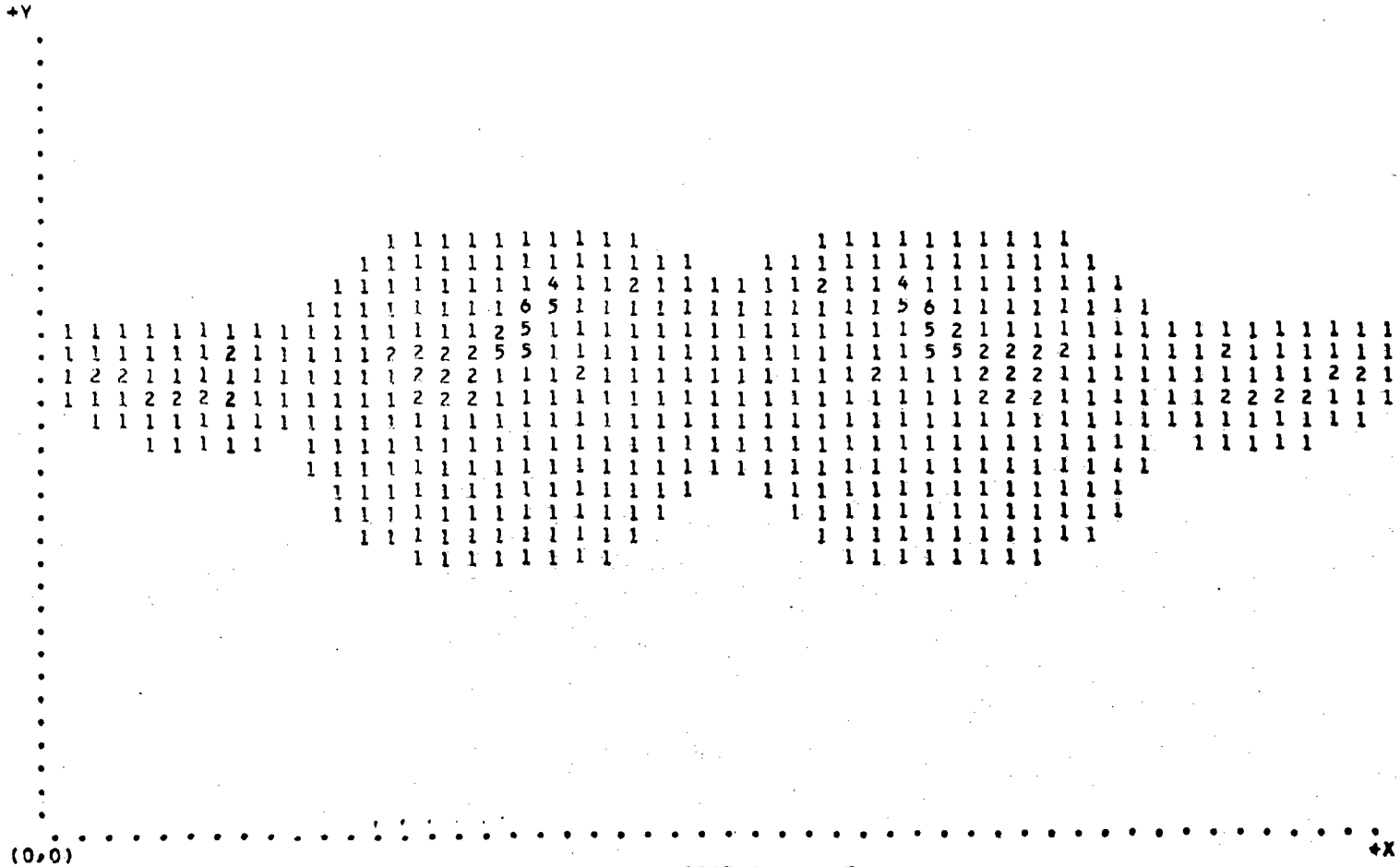


CROSS SECTION 34
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT

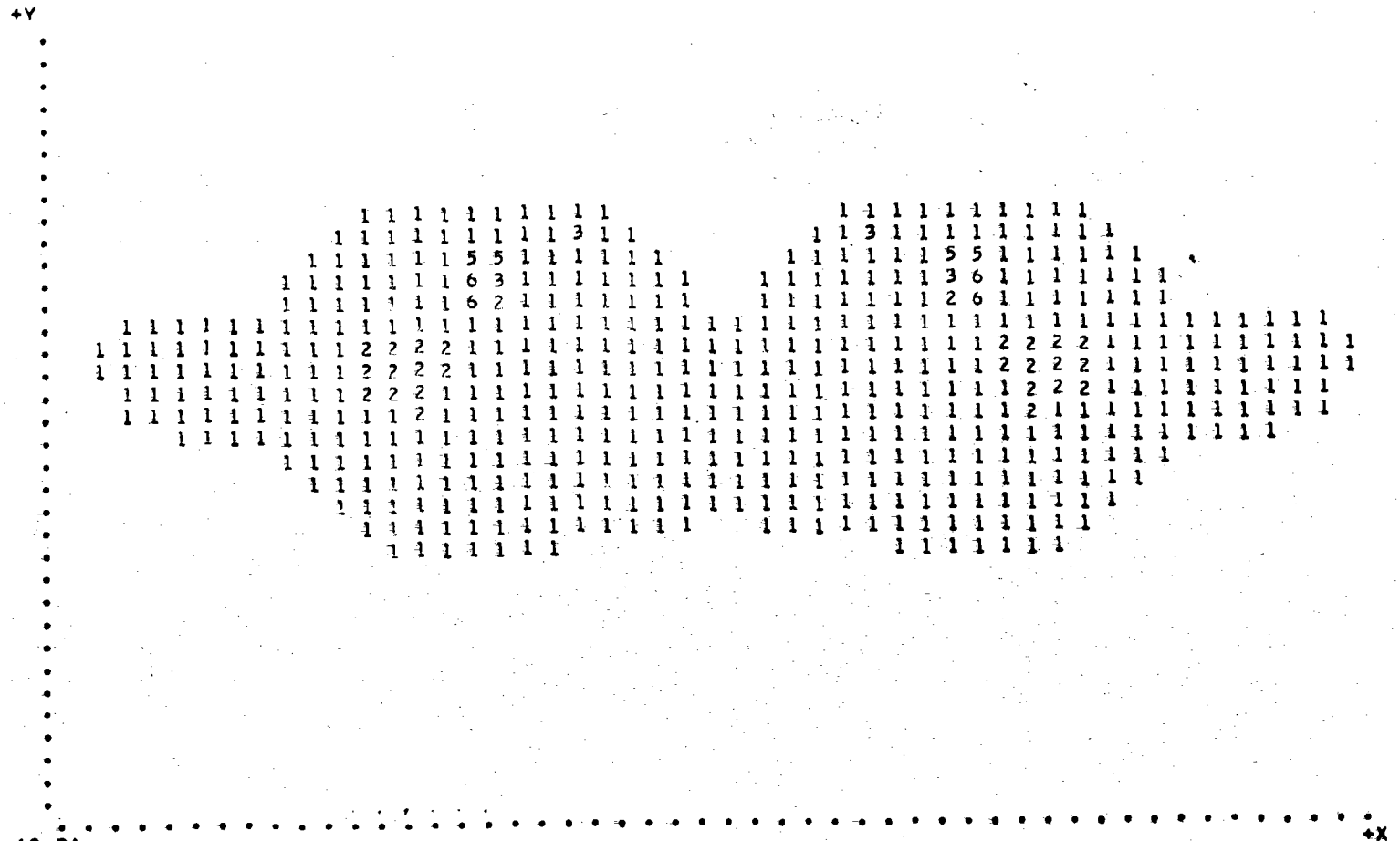


CROSS SECTION 35
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT

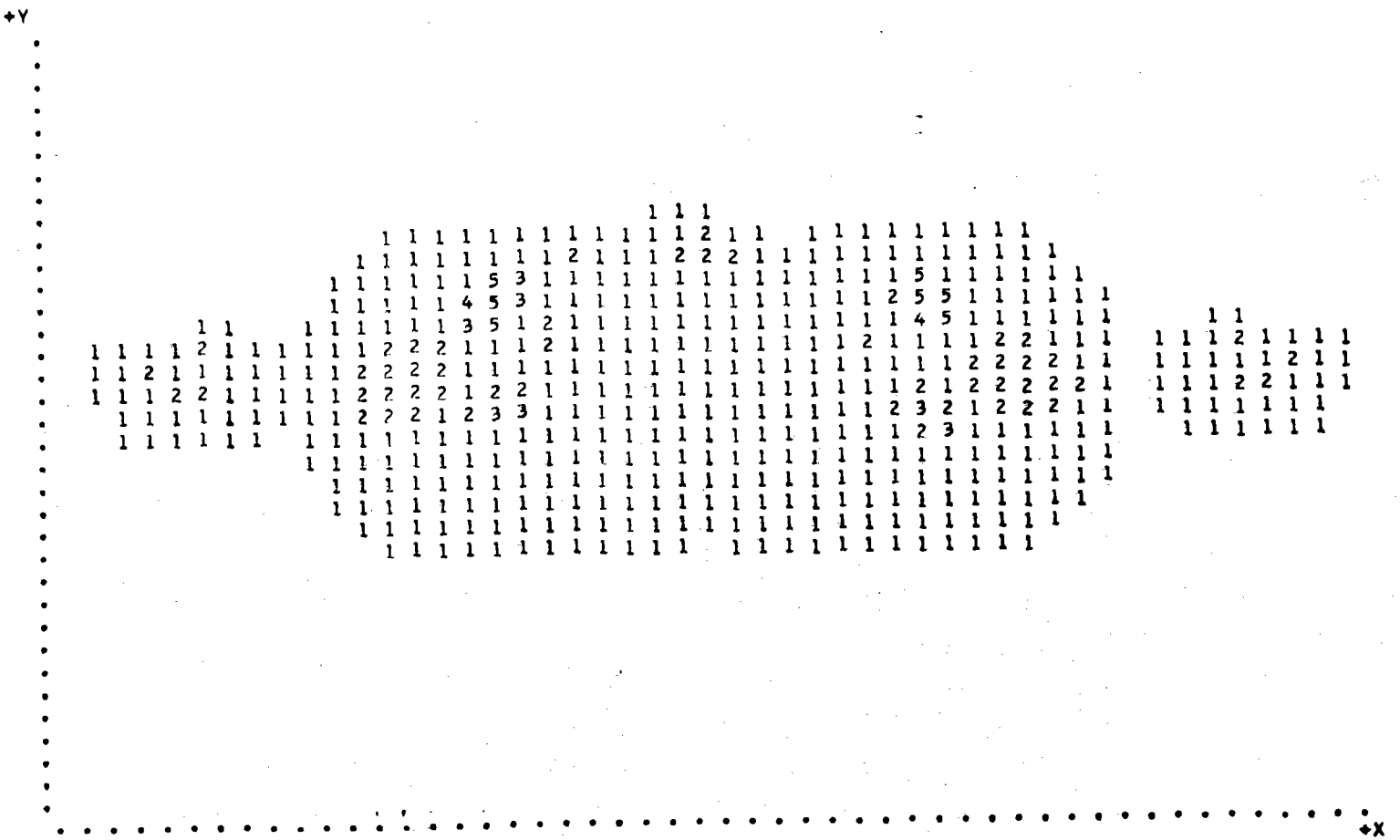
+Y
(0,0)
+X
CROSS SECTION 36
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



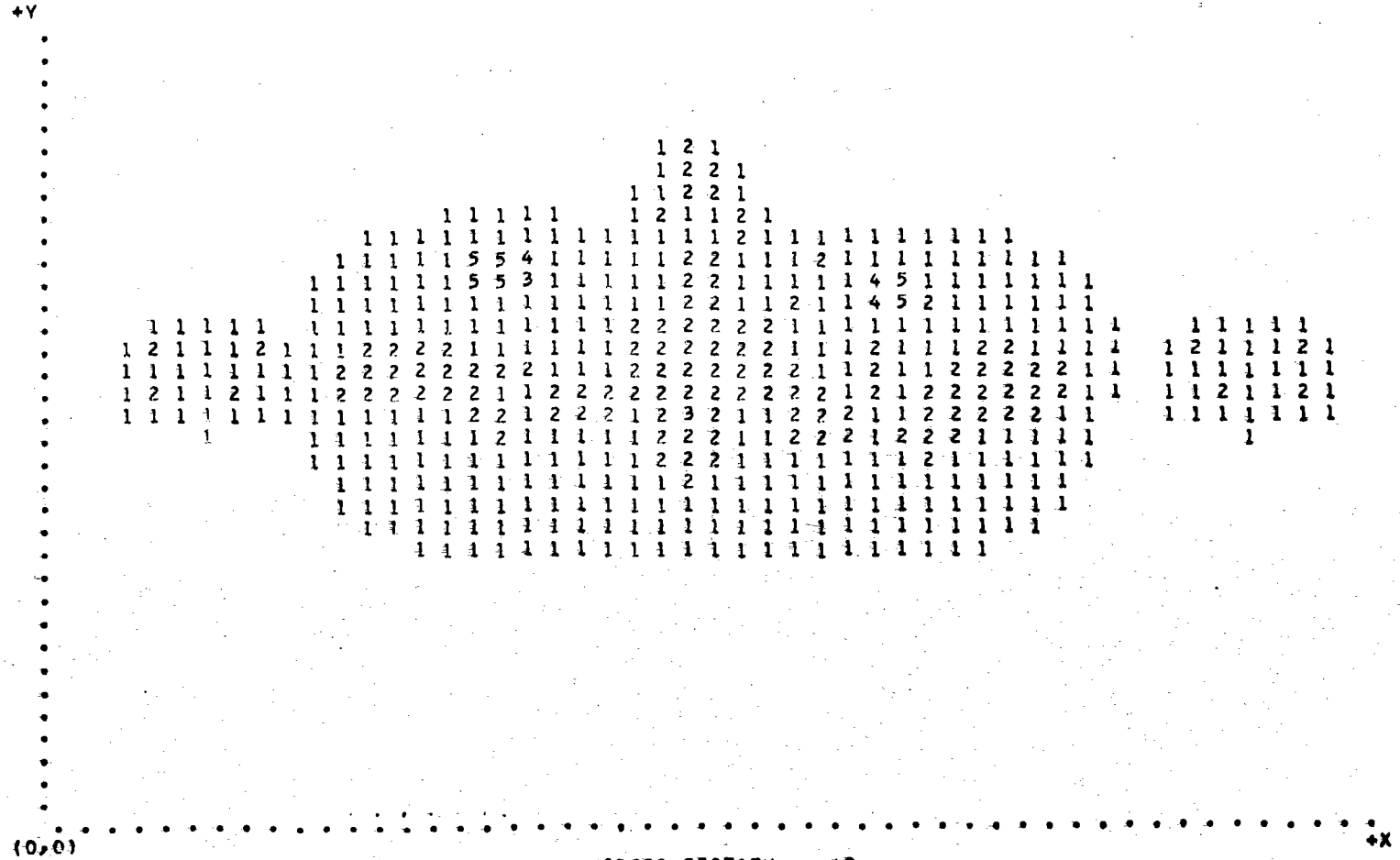
CROSS SECTION 37
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



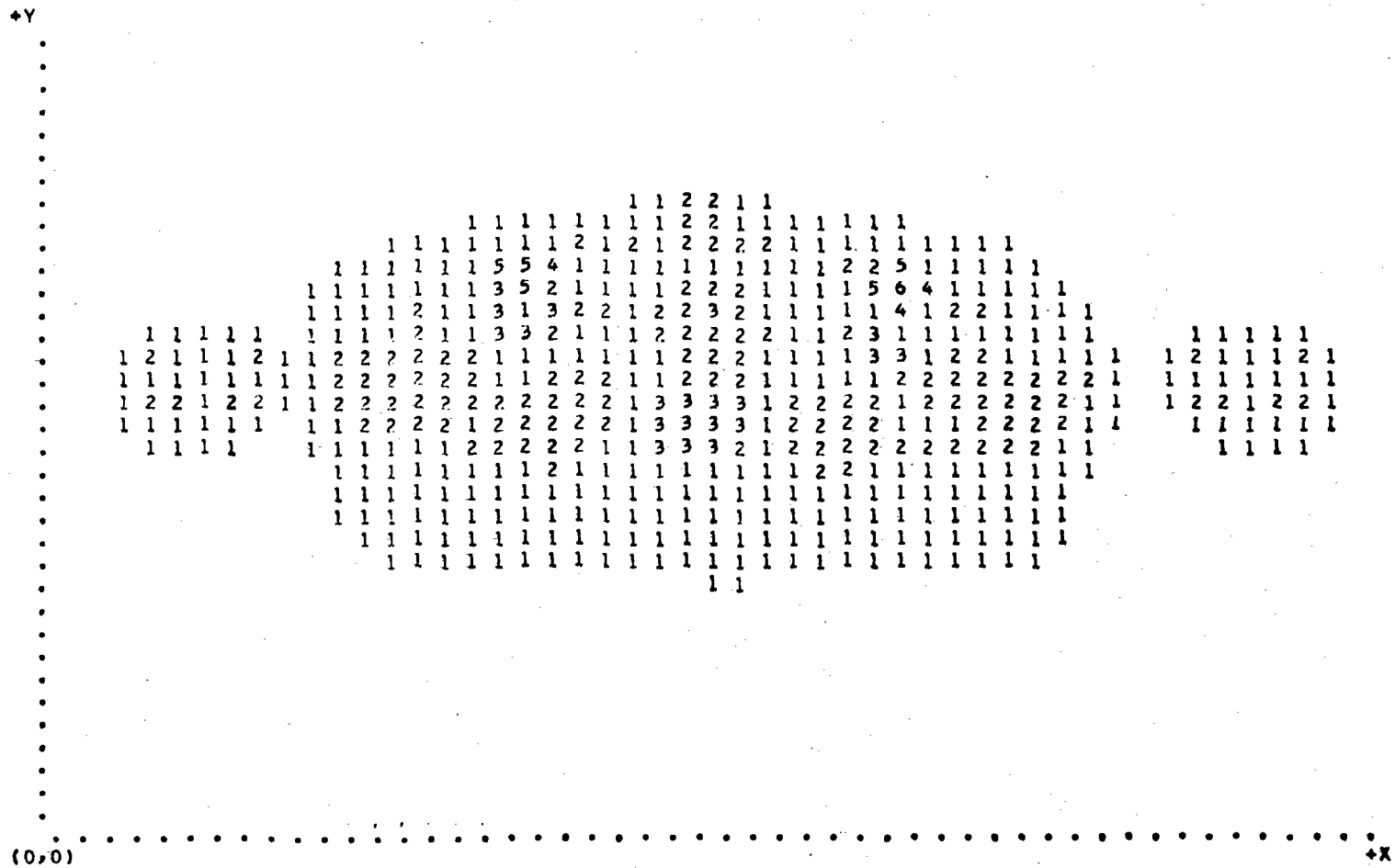
CROSS SECTION 38
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



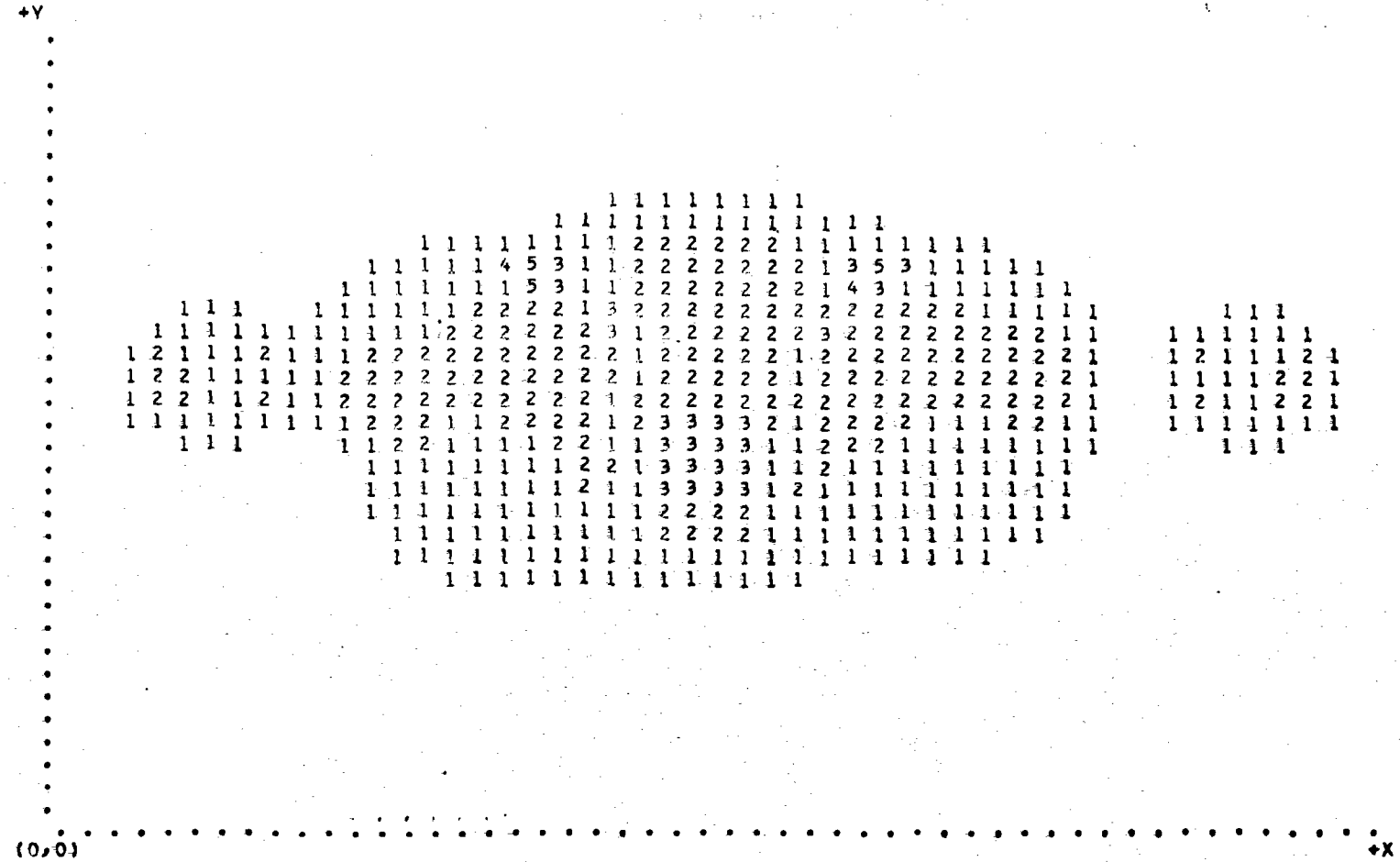
CROSS SECTION 39
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



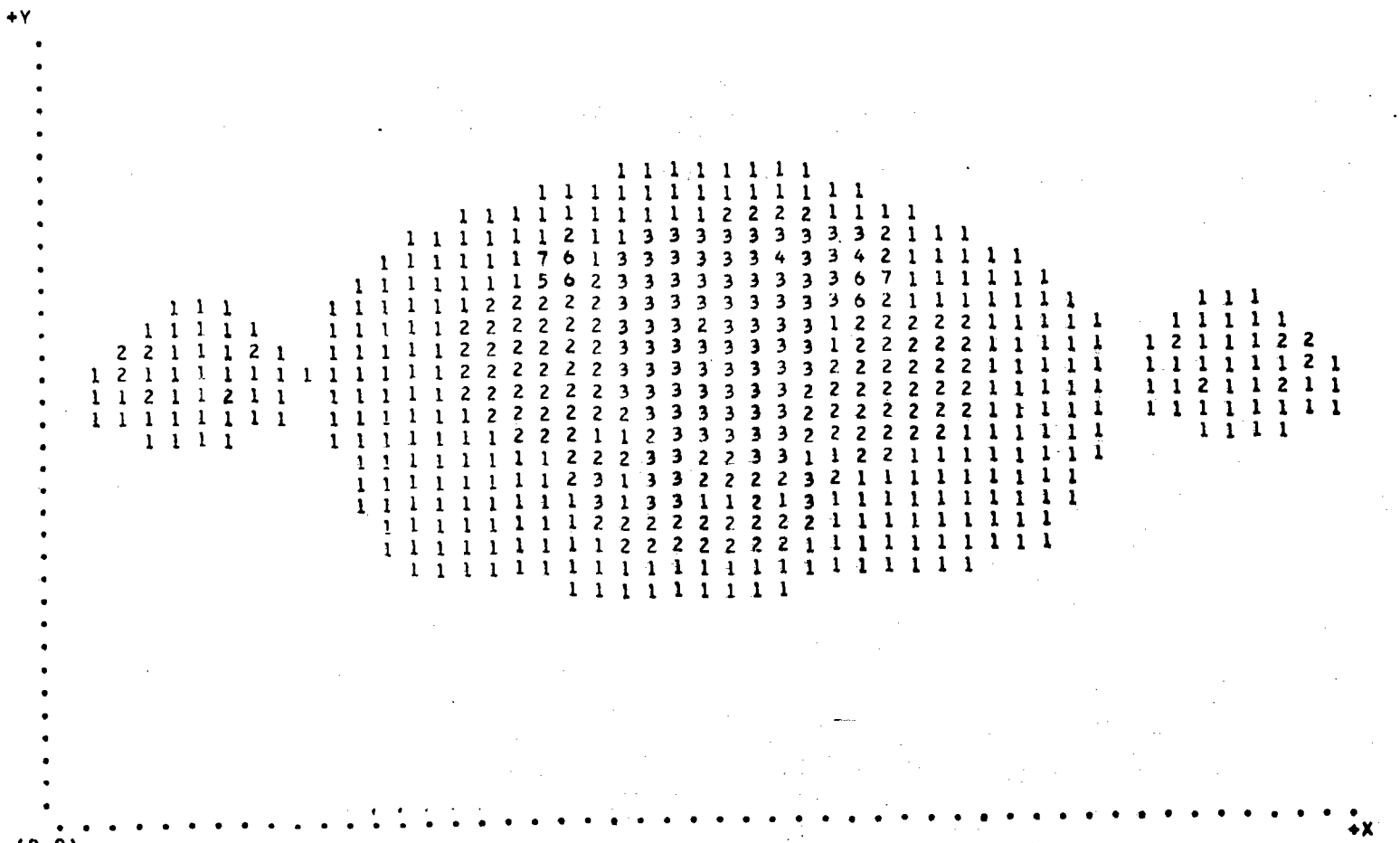
CROSS SECTION 40
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



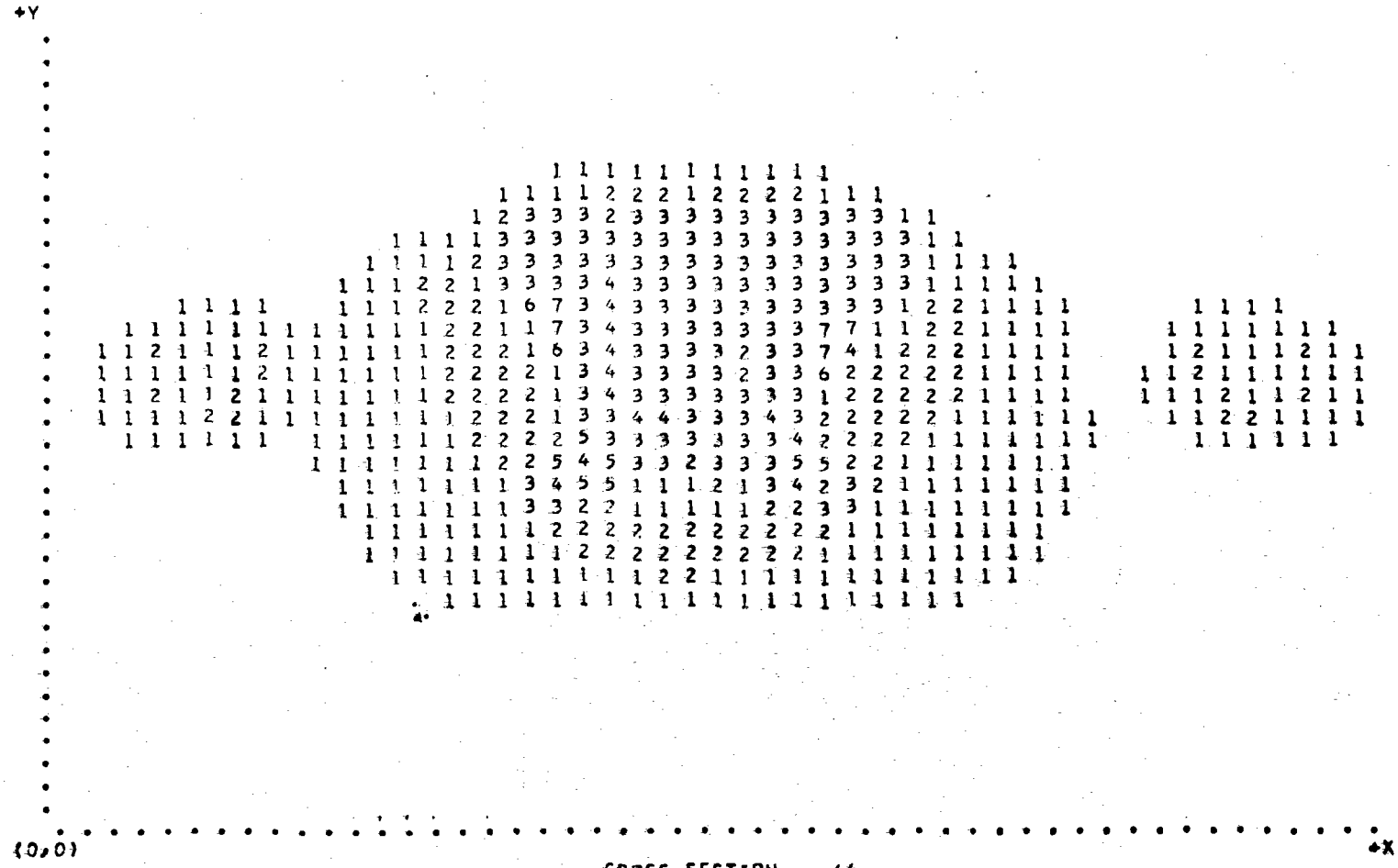
CROSS SECTION 41
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



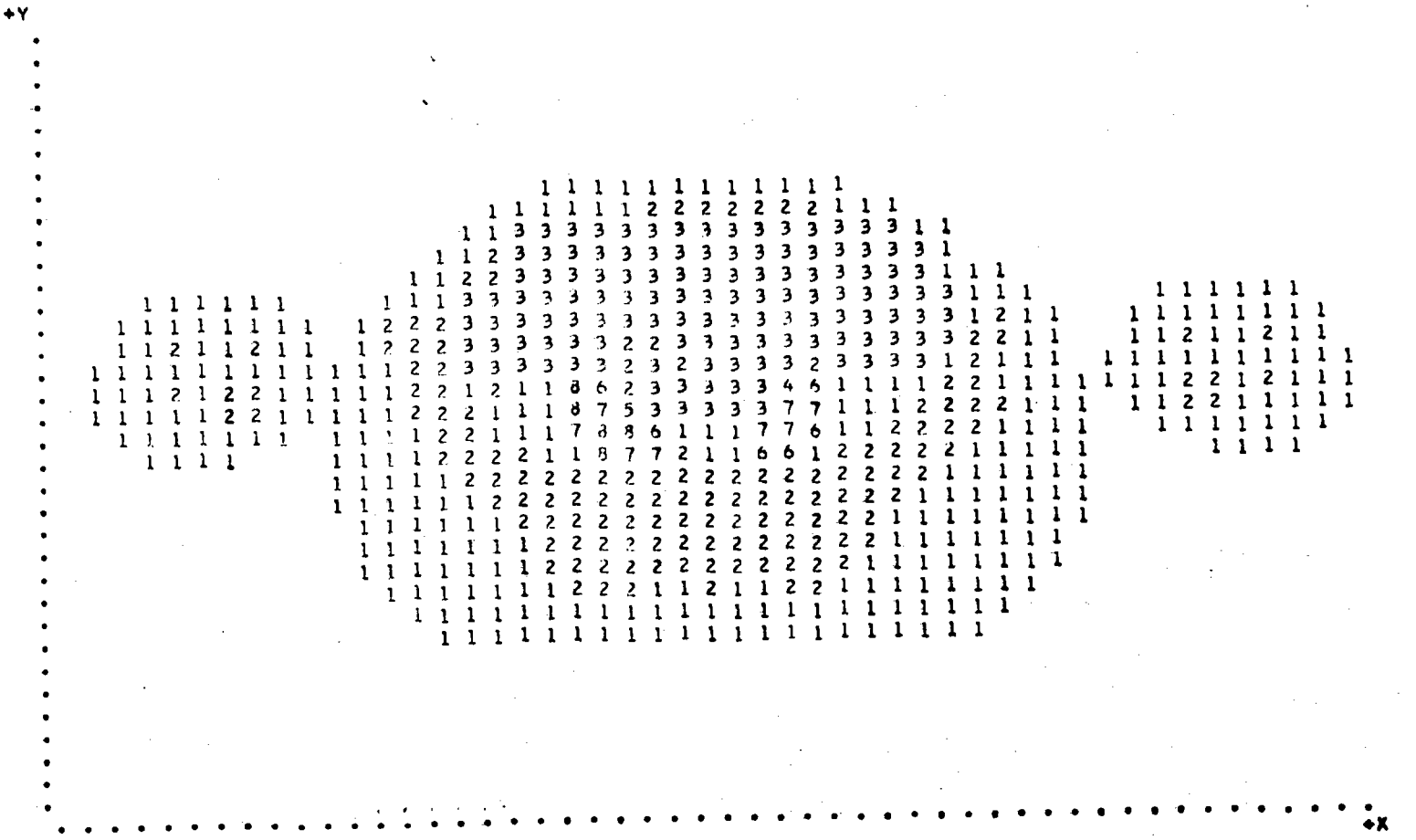
CROSS SECTION 42
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



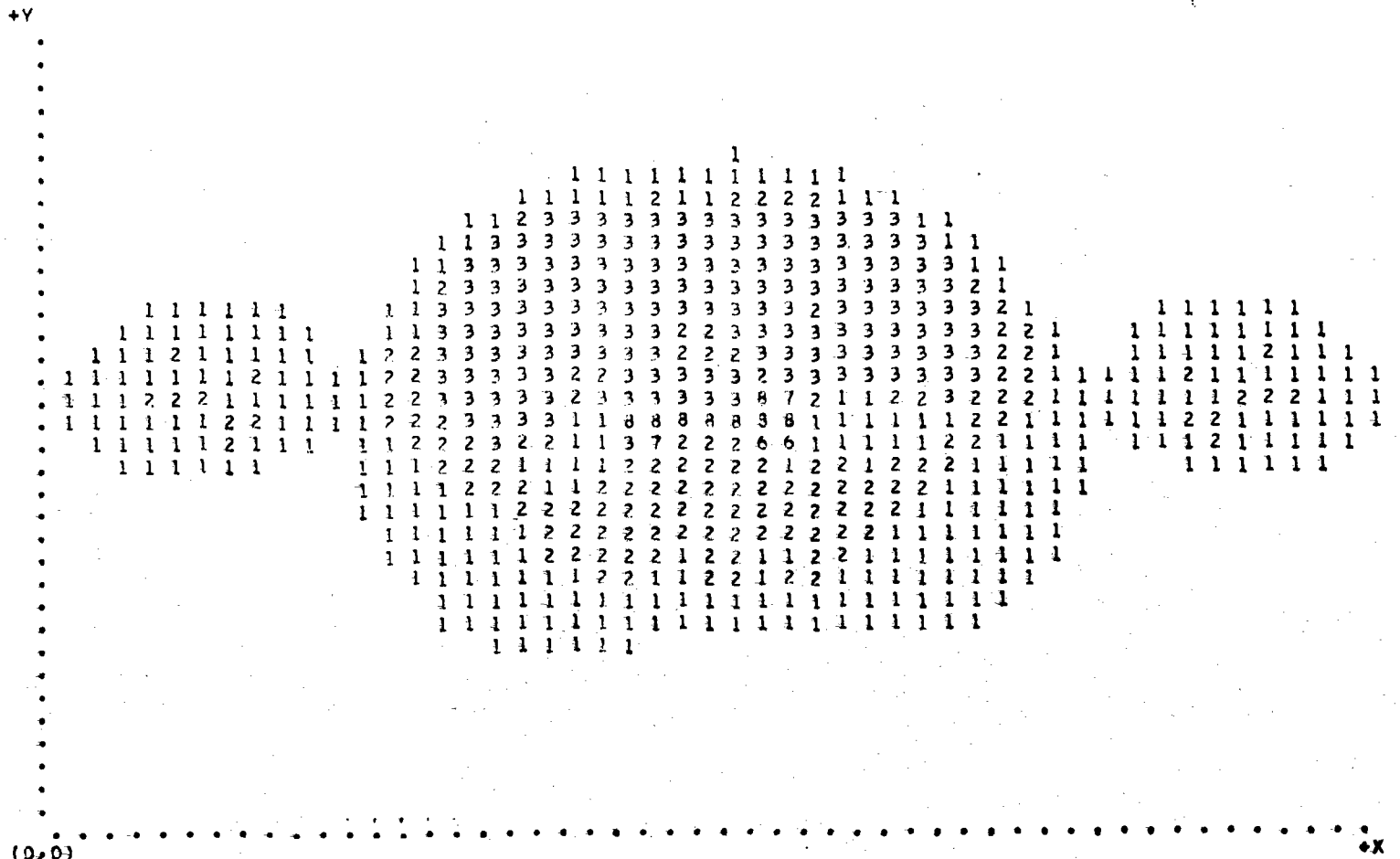
CROSS SECTION 43
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



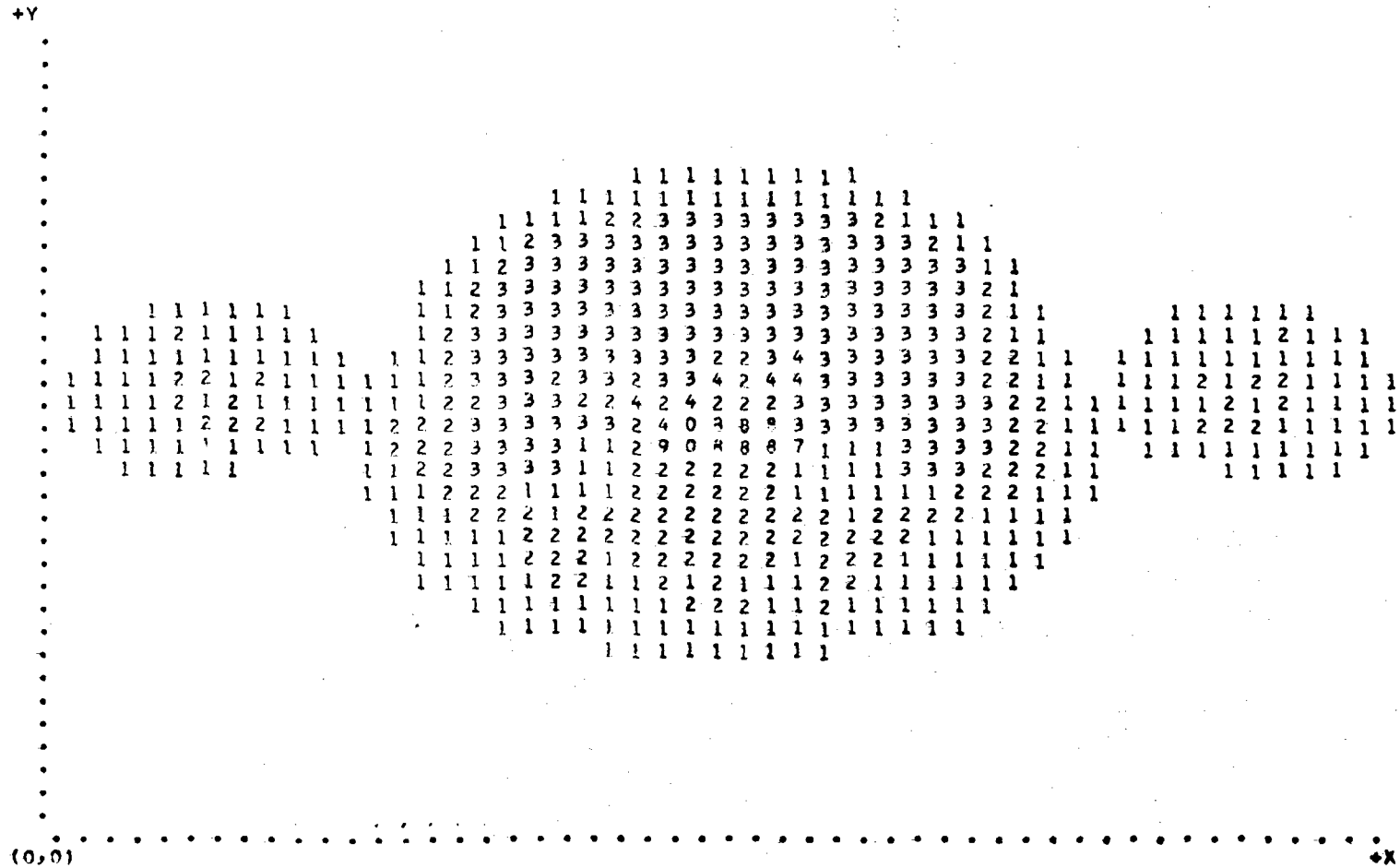
CROSS SECTION 44
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



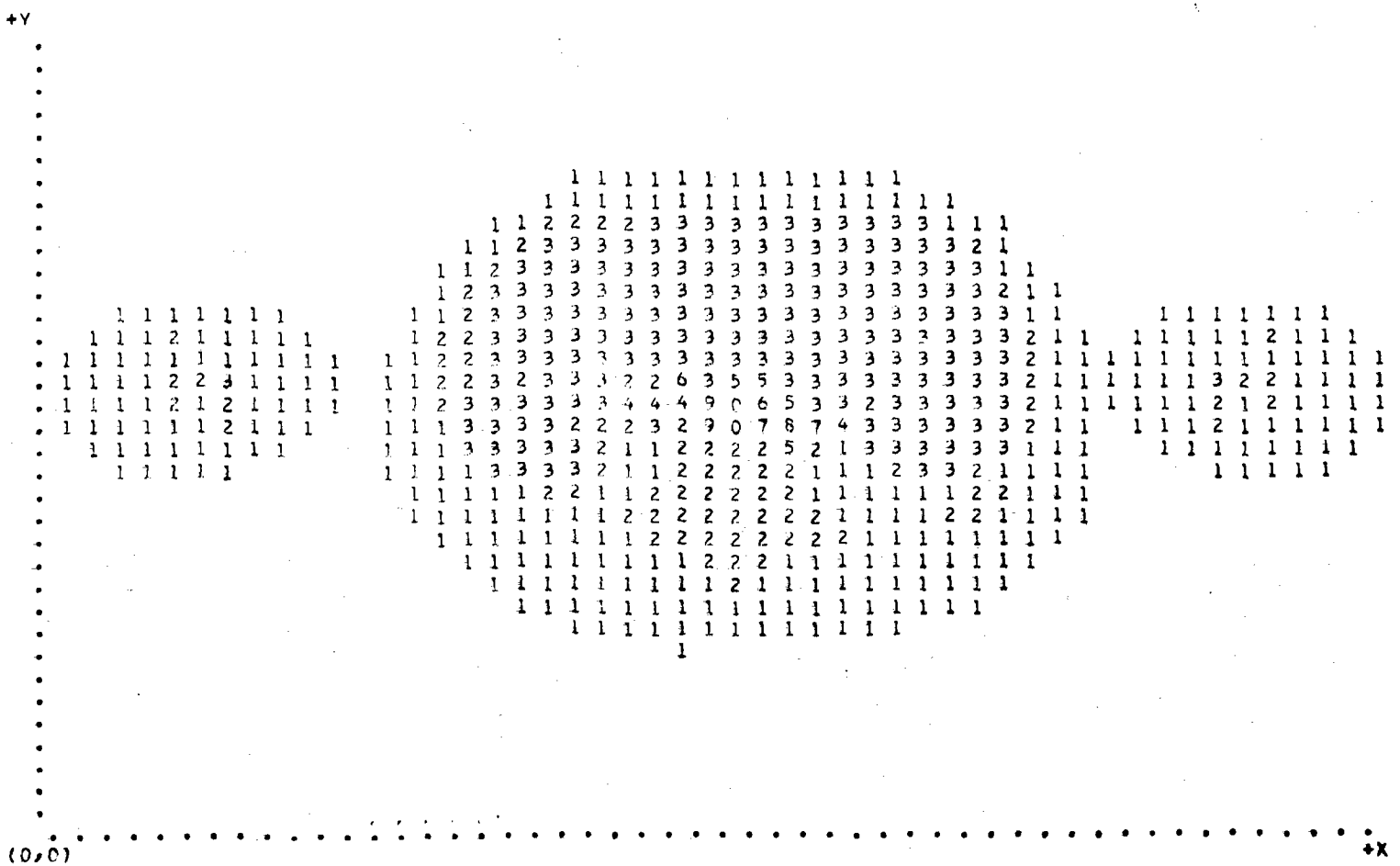
CROSS SECTION 45
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



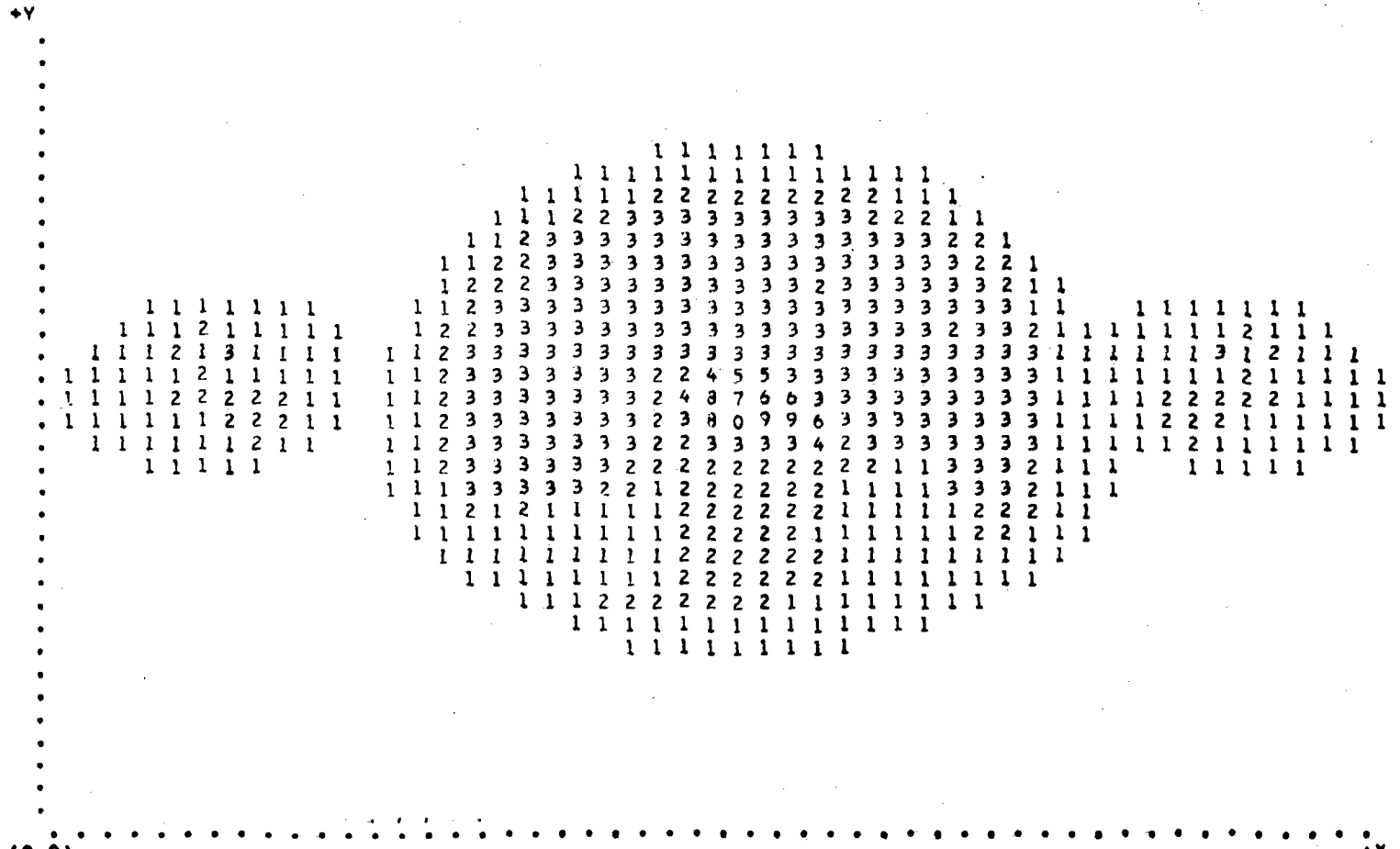
CROSS SECTION 46
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



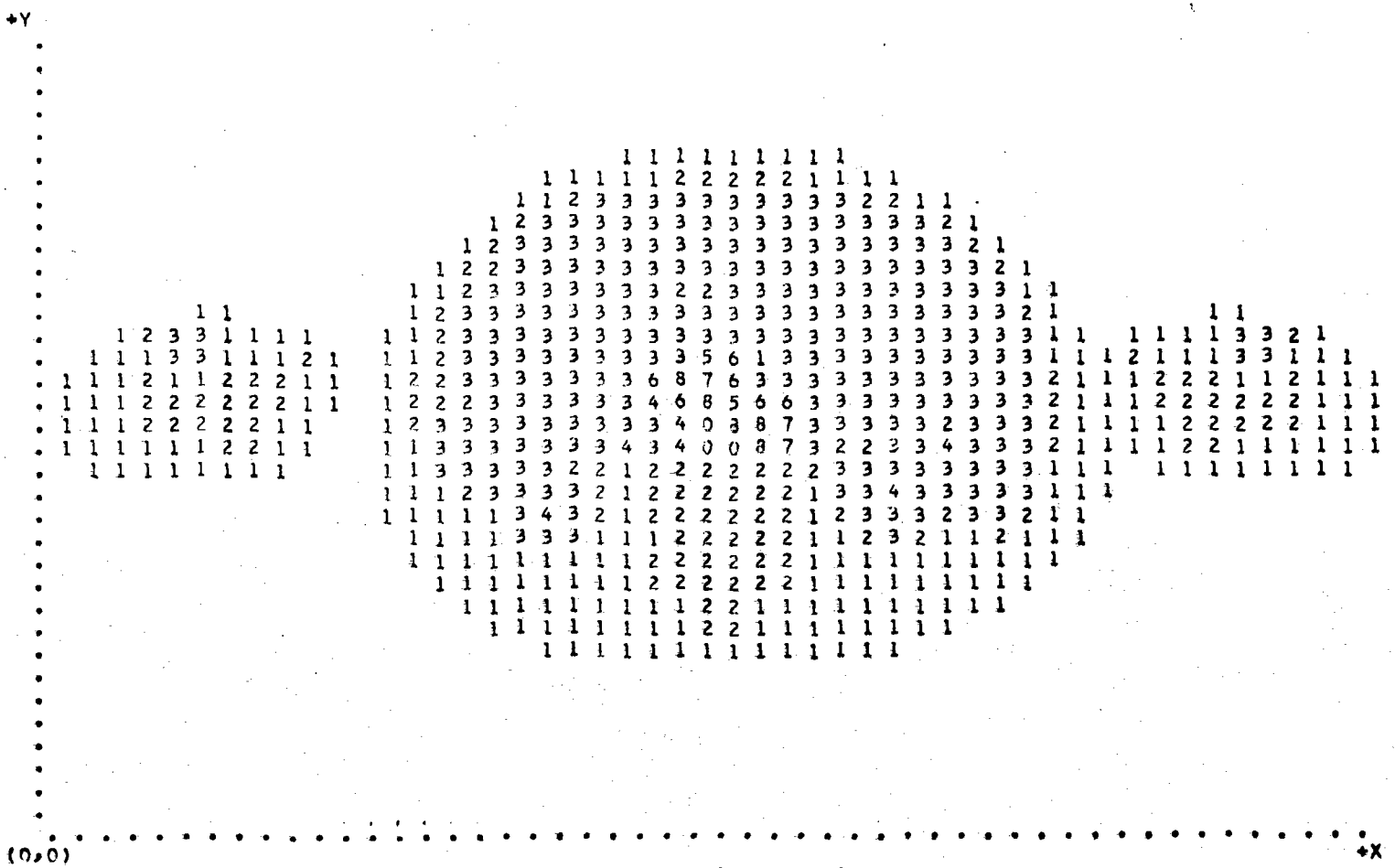
CROSS SECTION 47
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



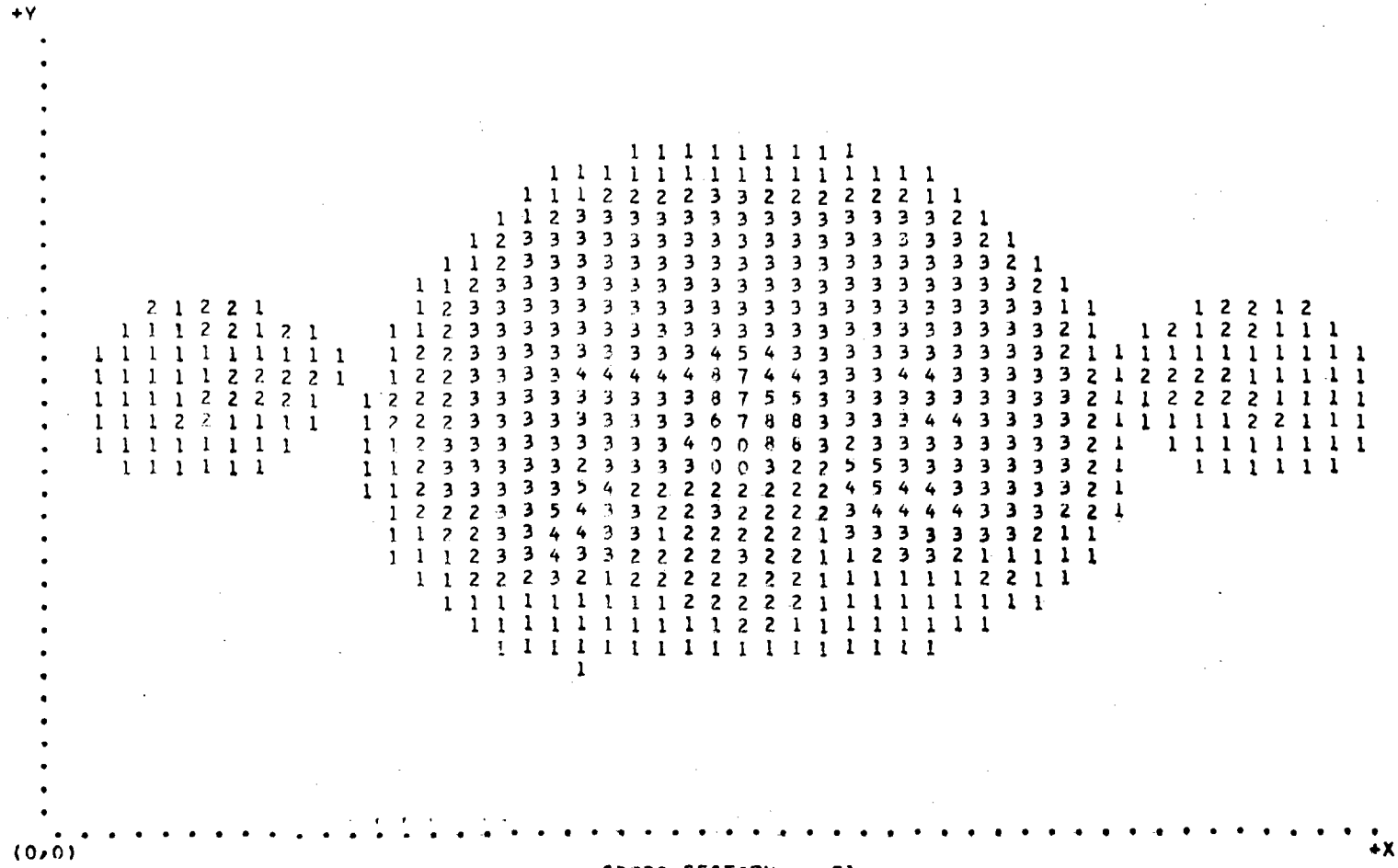
CROSS SECTION 48
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



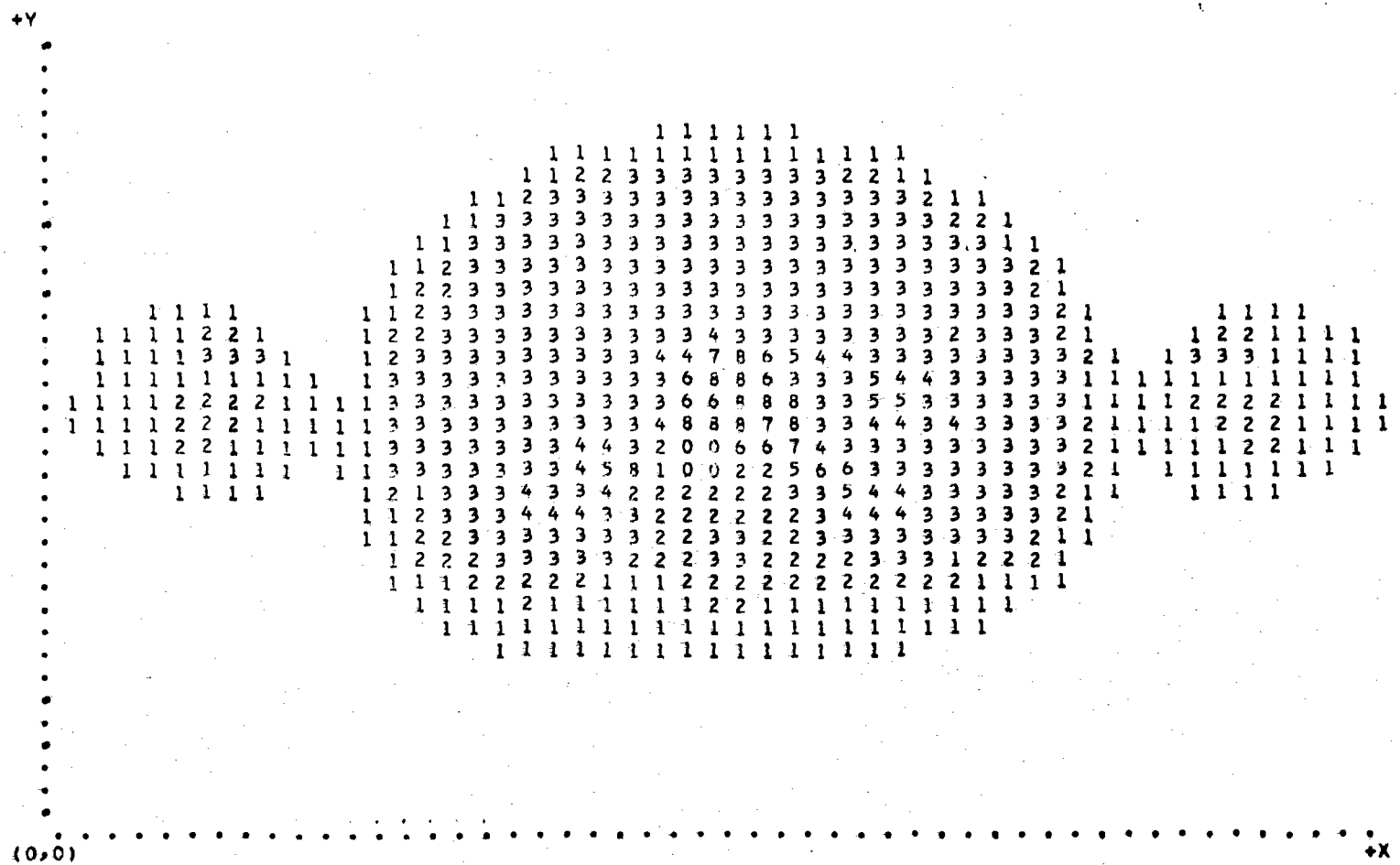
CROSS SECTION 49
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



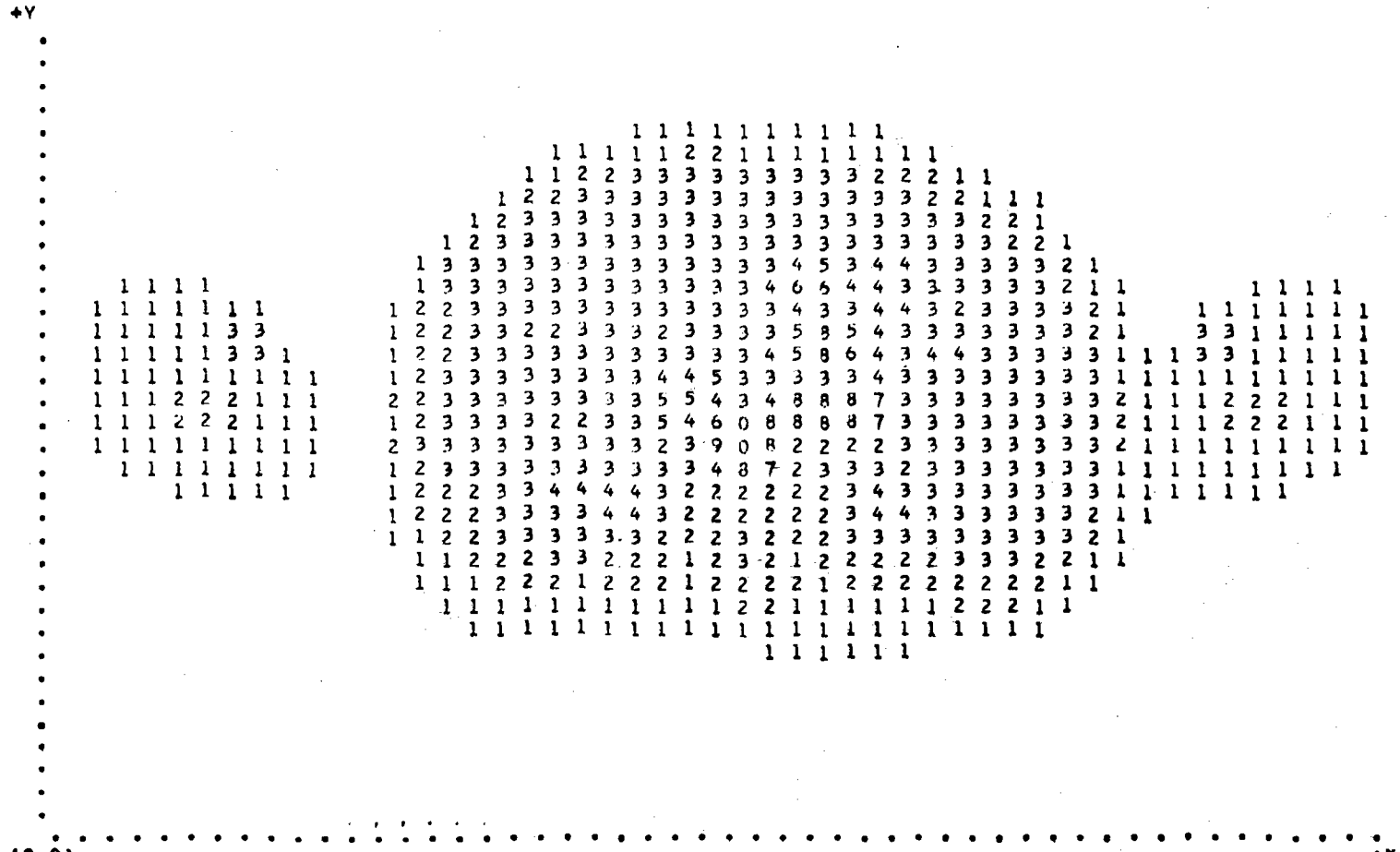
CROSS SECTION 50
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



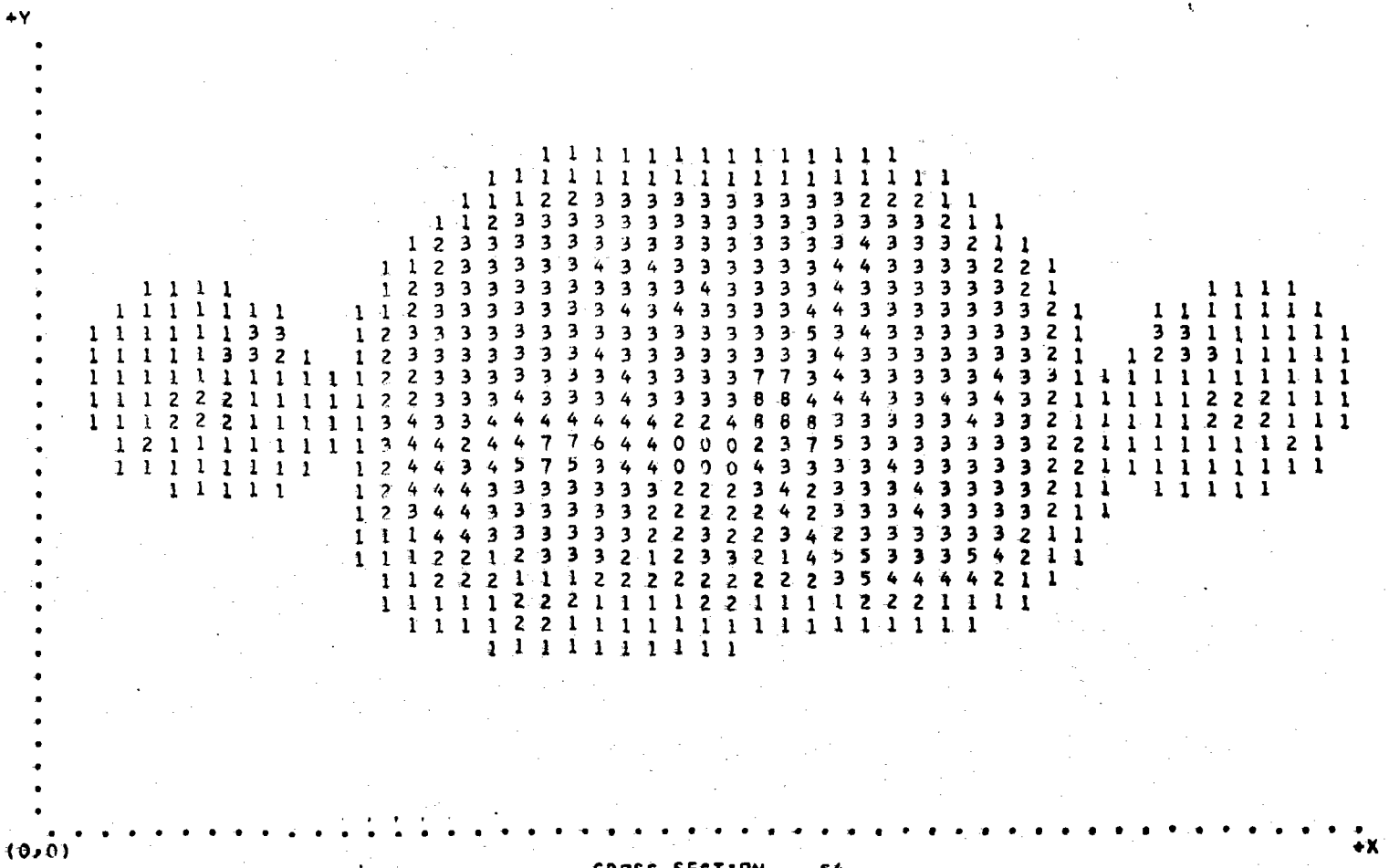
CROSS SECTION 51
AVERAGE DOCTURS EVALUATION 1 HOUR ASSESSMENT



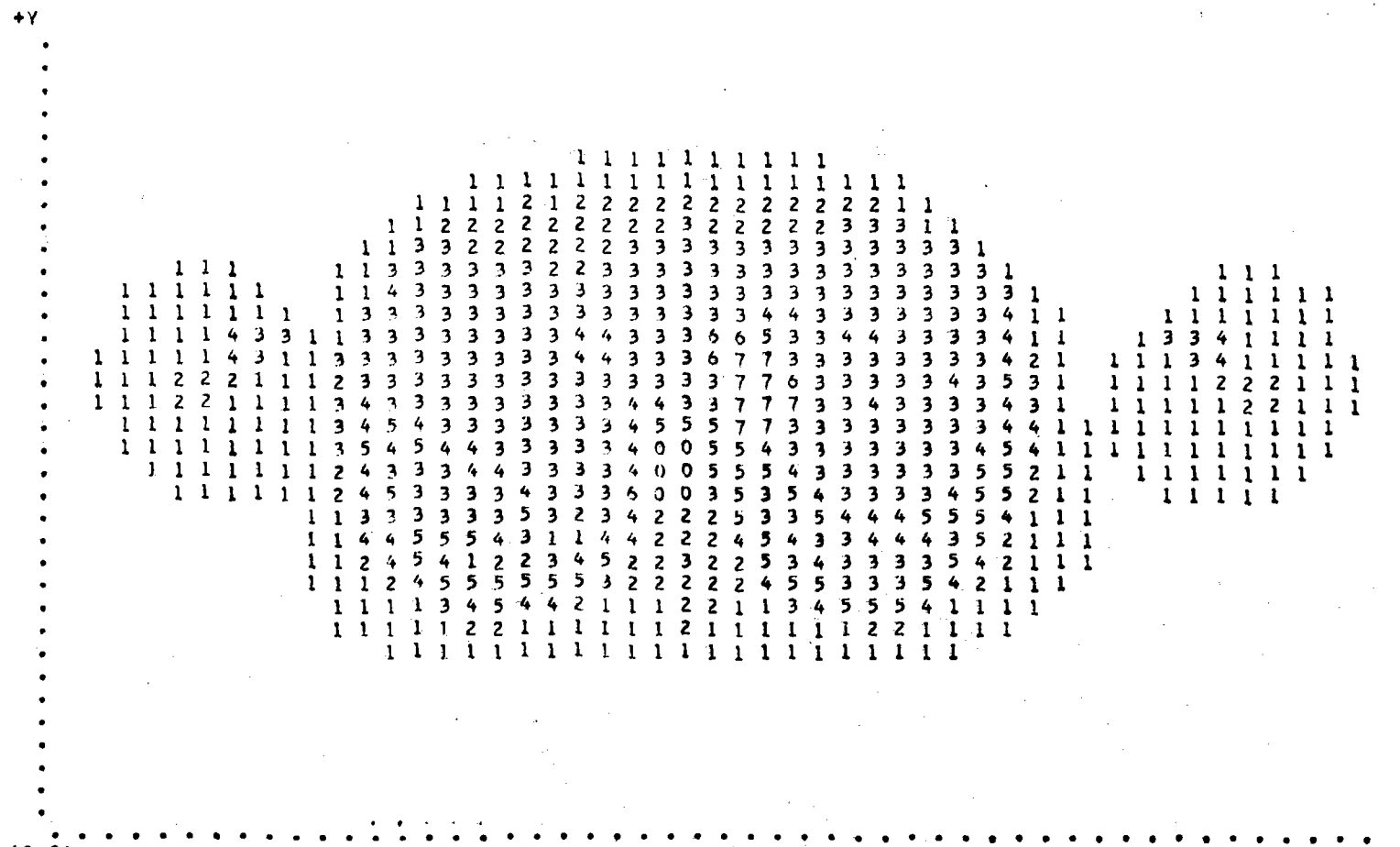
CROSS SECTION 52
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



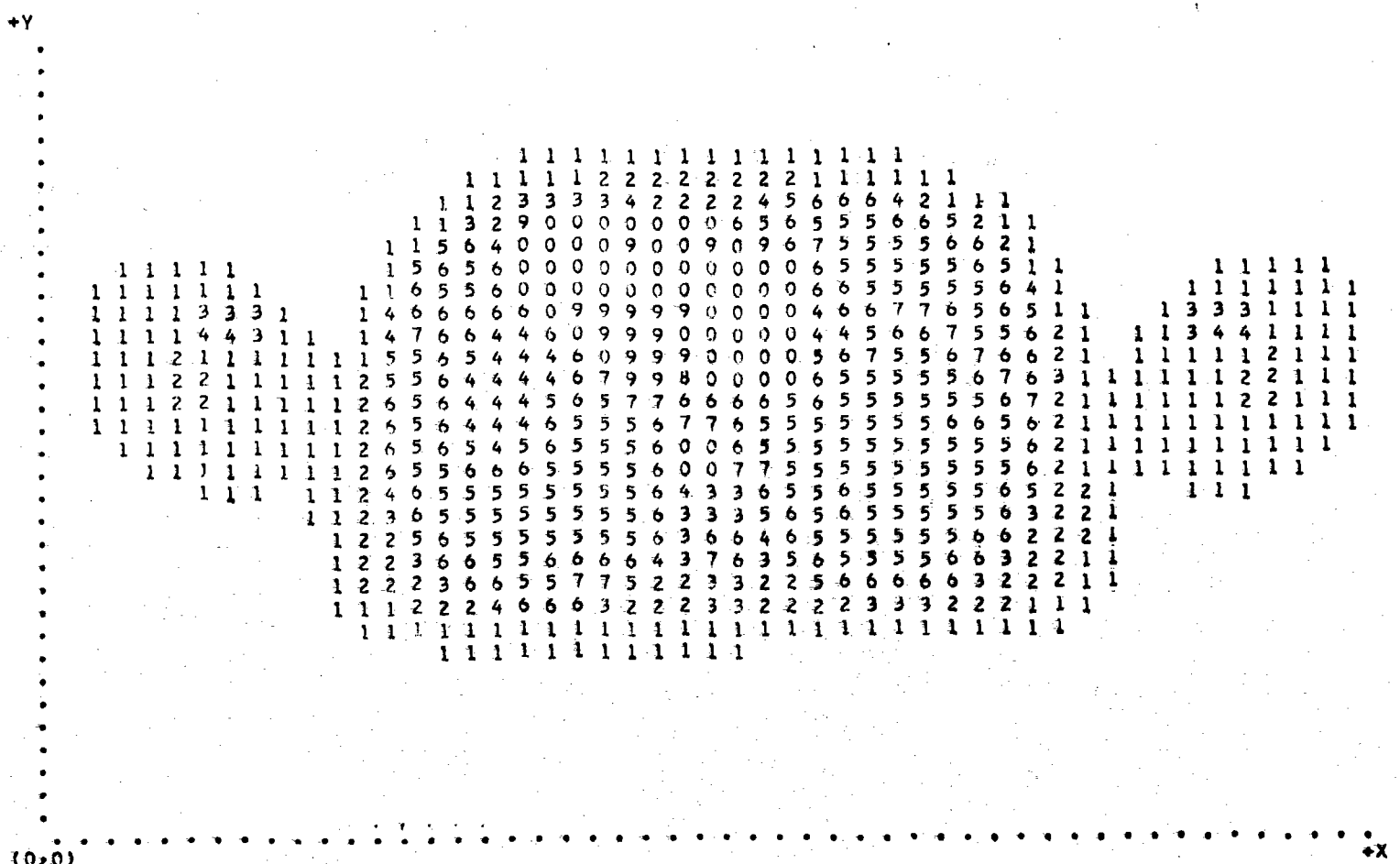
CROSS SECTION 53
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



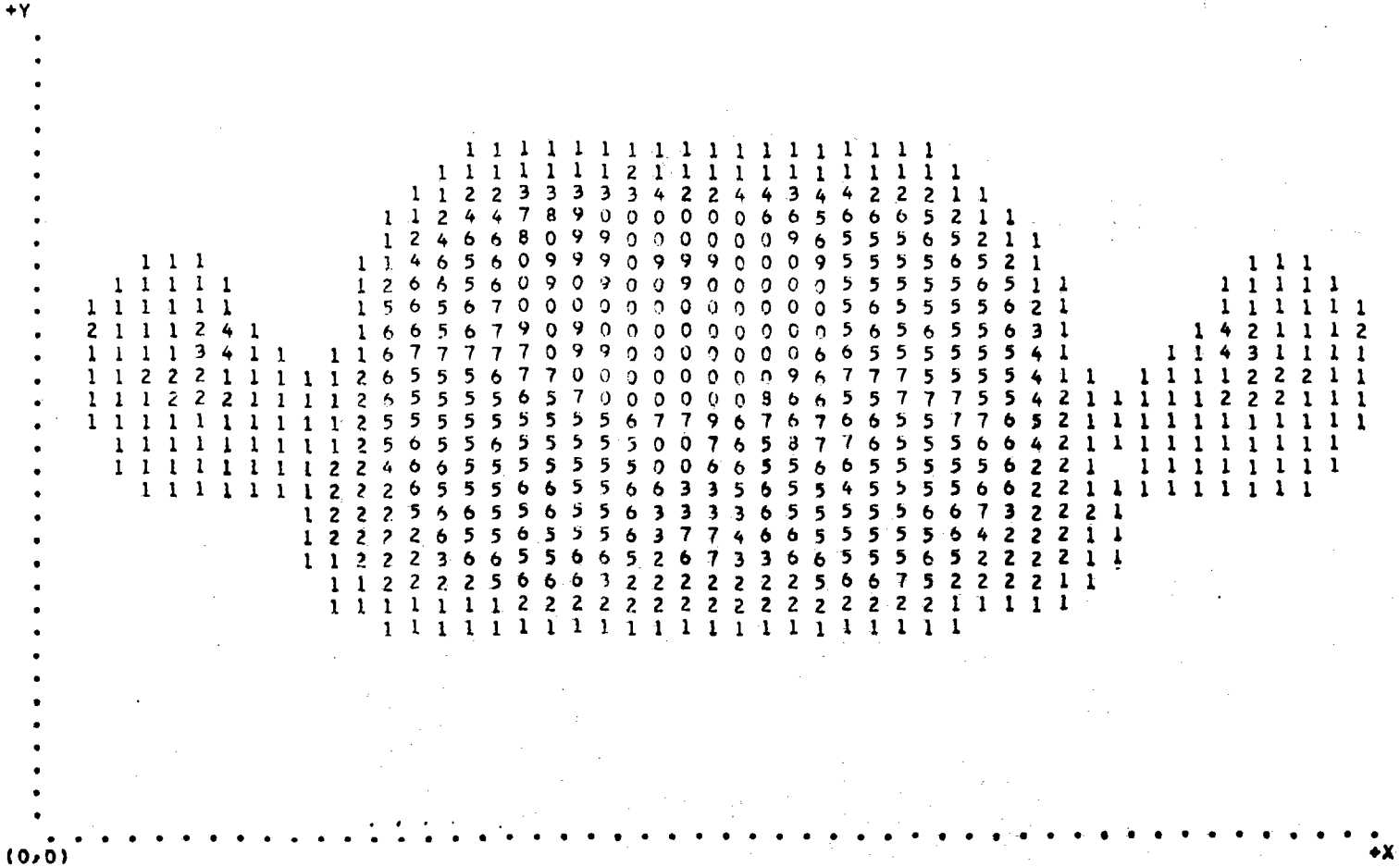
CROSS SECTION 54
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



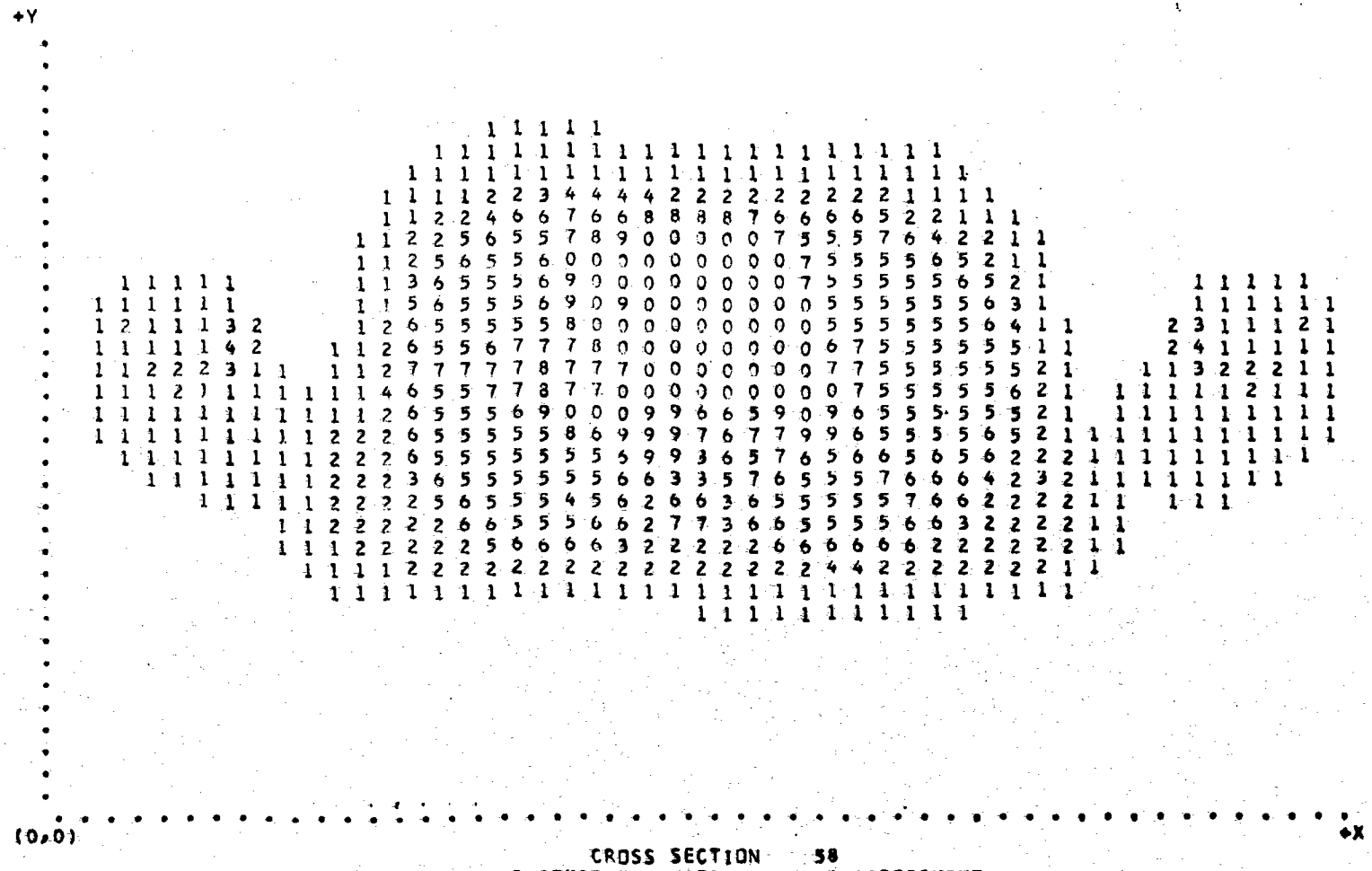
CROSS SECTION 55
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



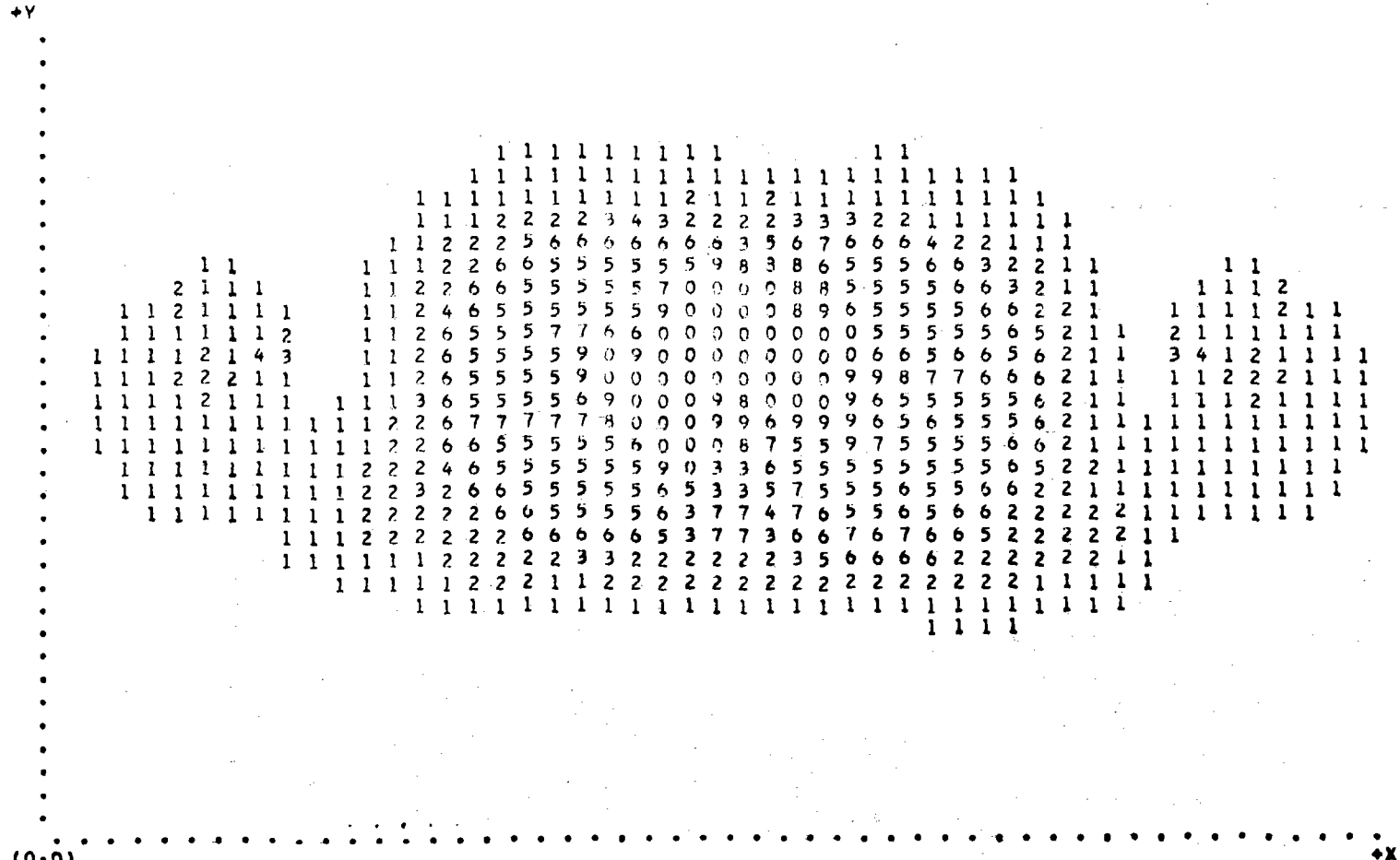
CROSS SECTION 56
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



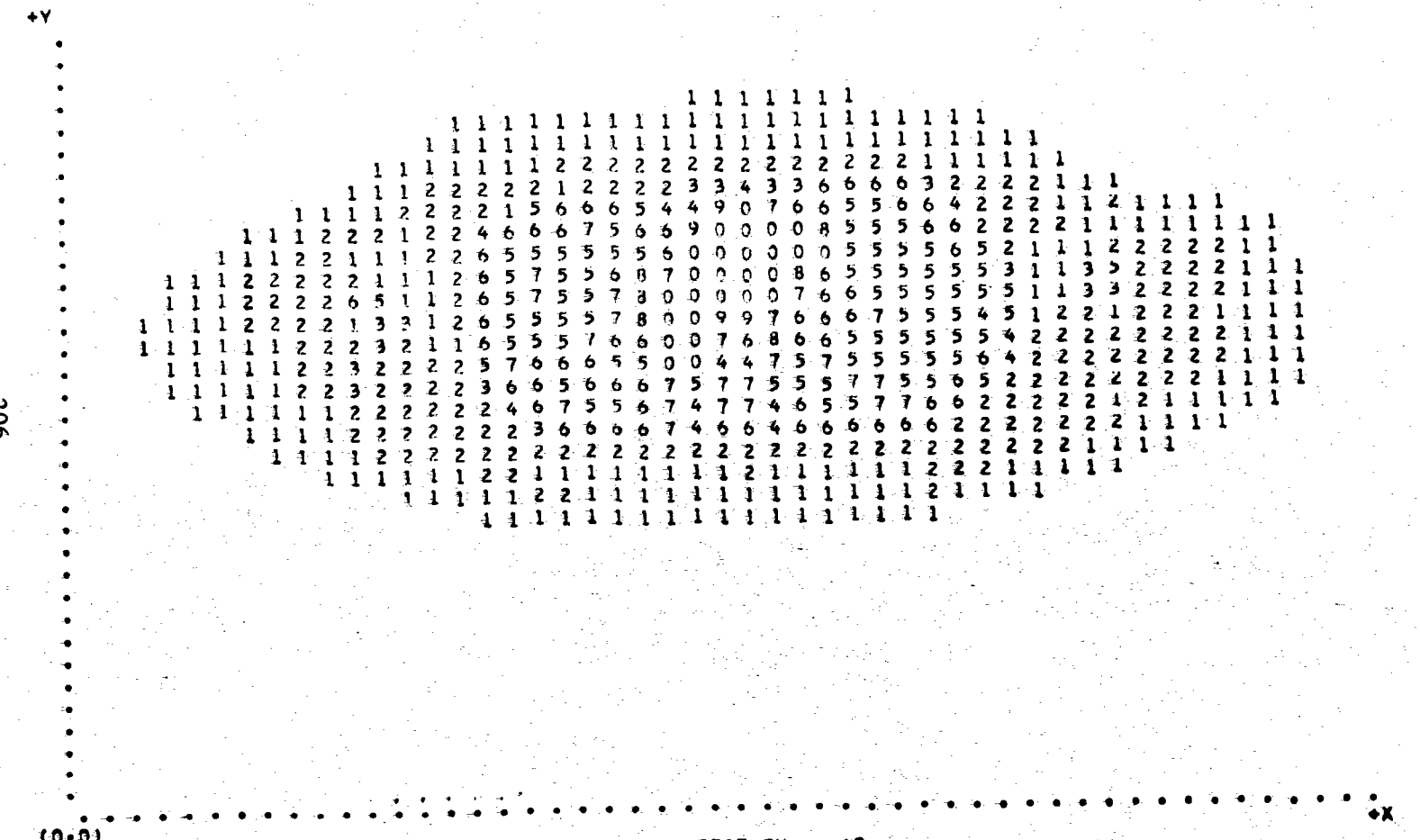
CROSS SECTION 57
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



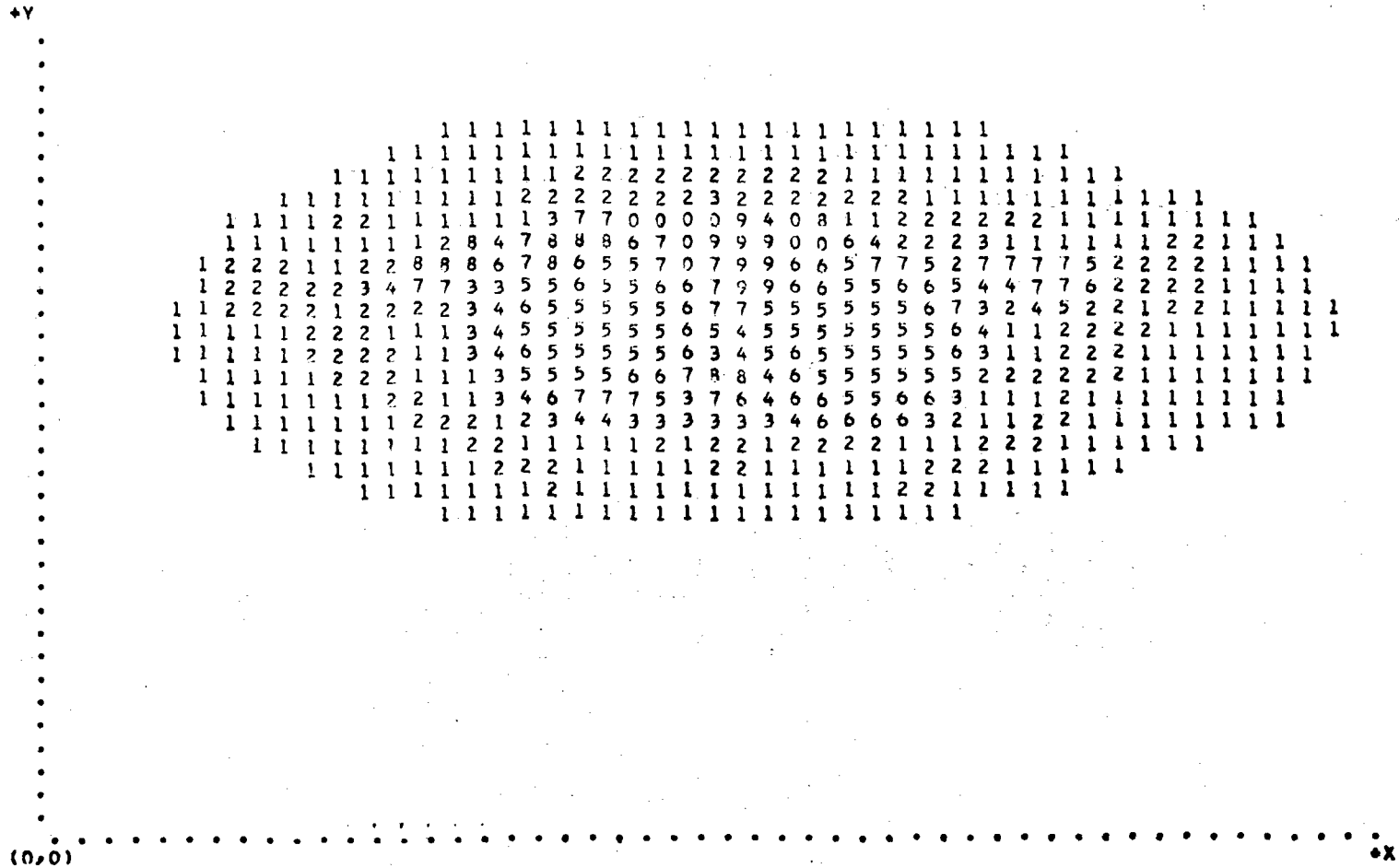
CROSS SECTION 58
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



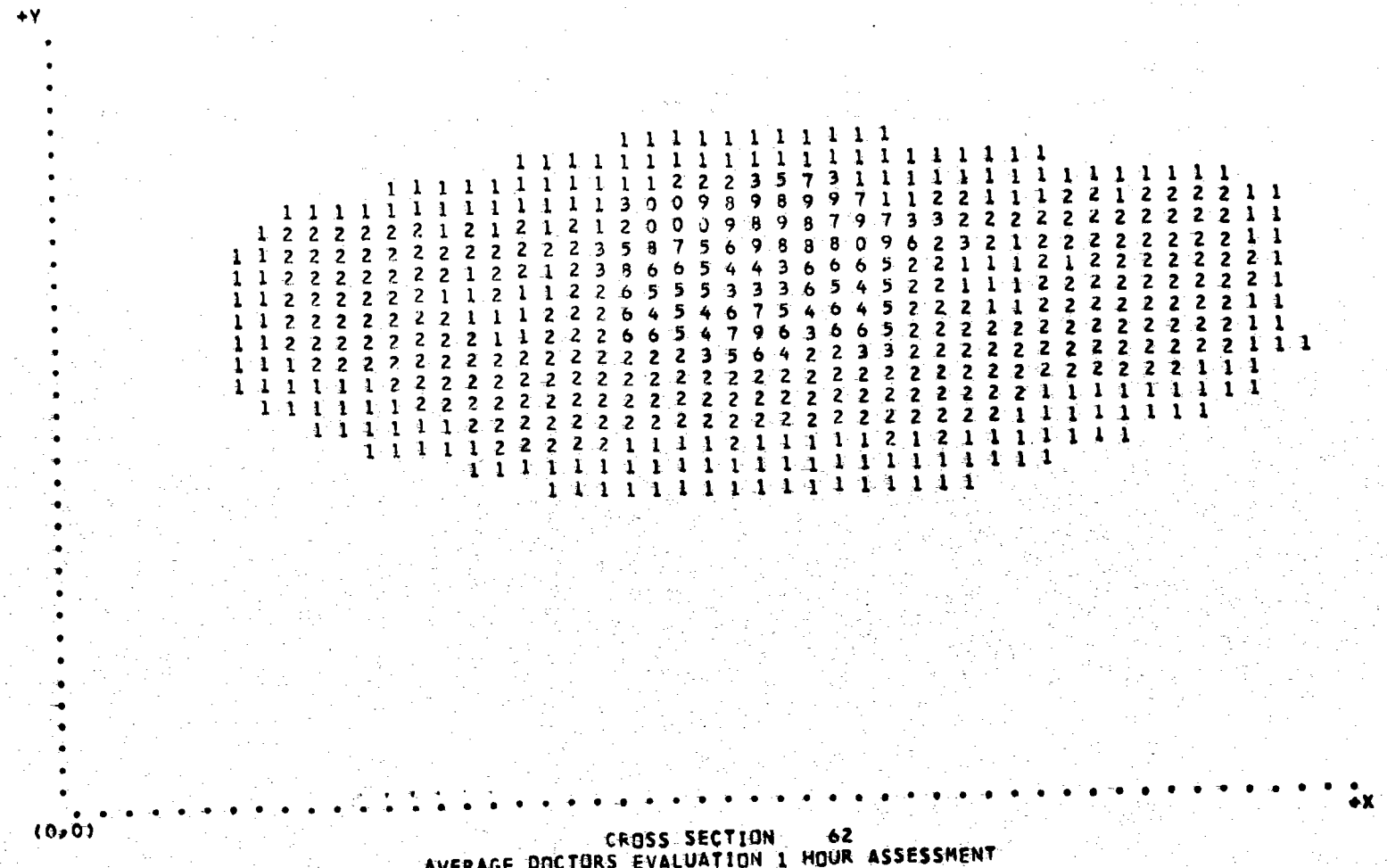
CROSS SECTION 59
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



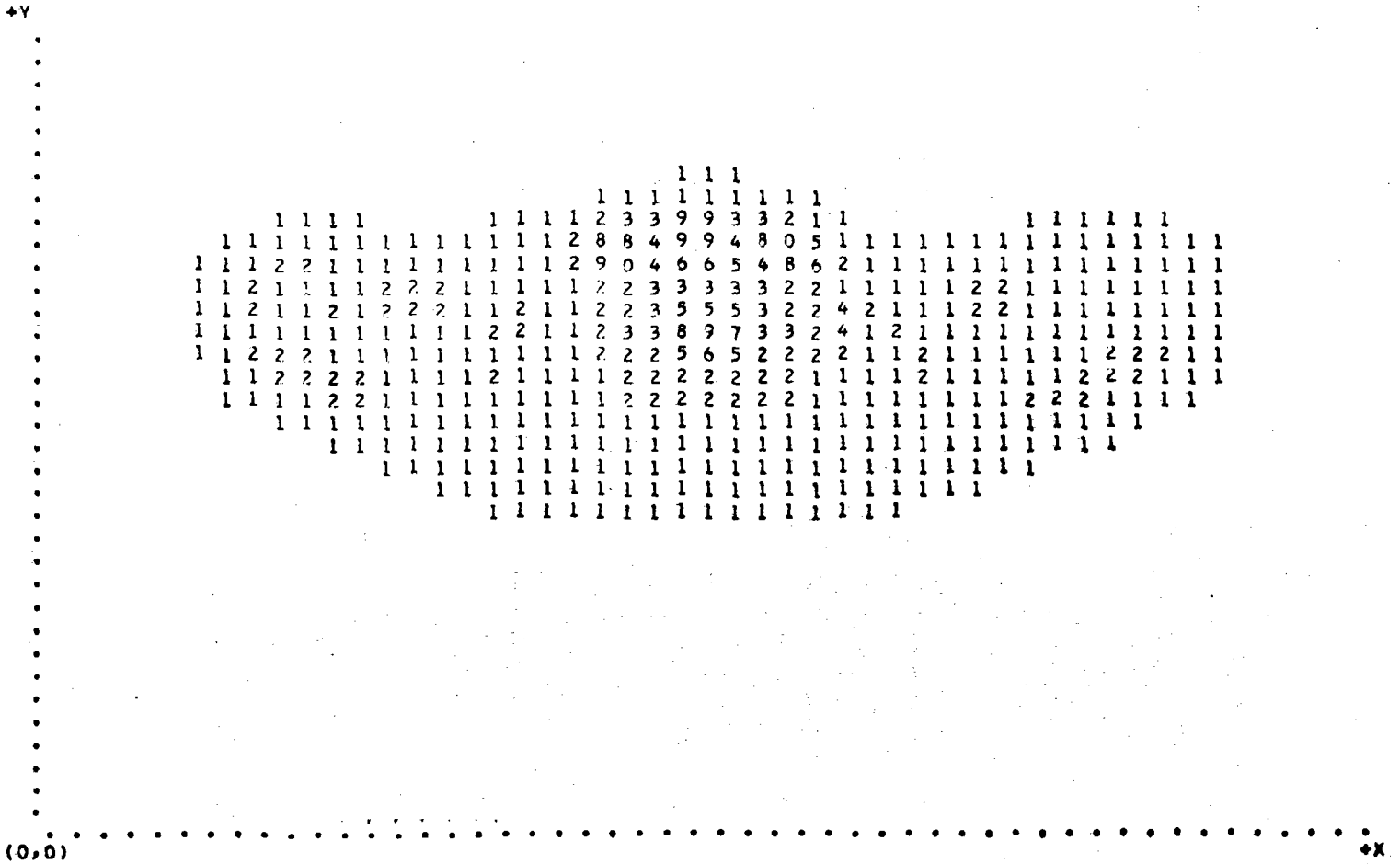
CROSS SECTION 60
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



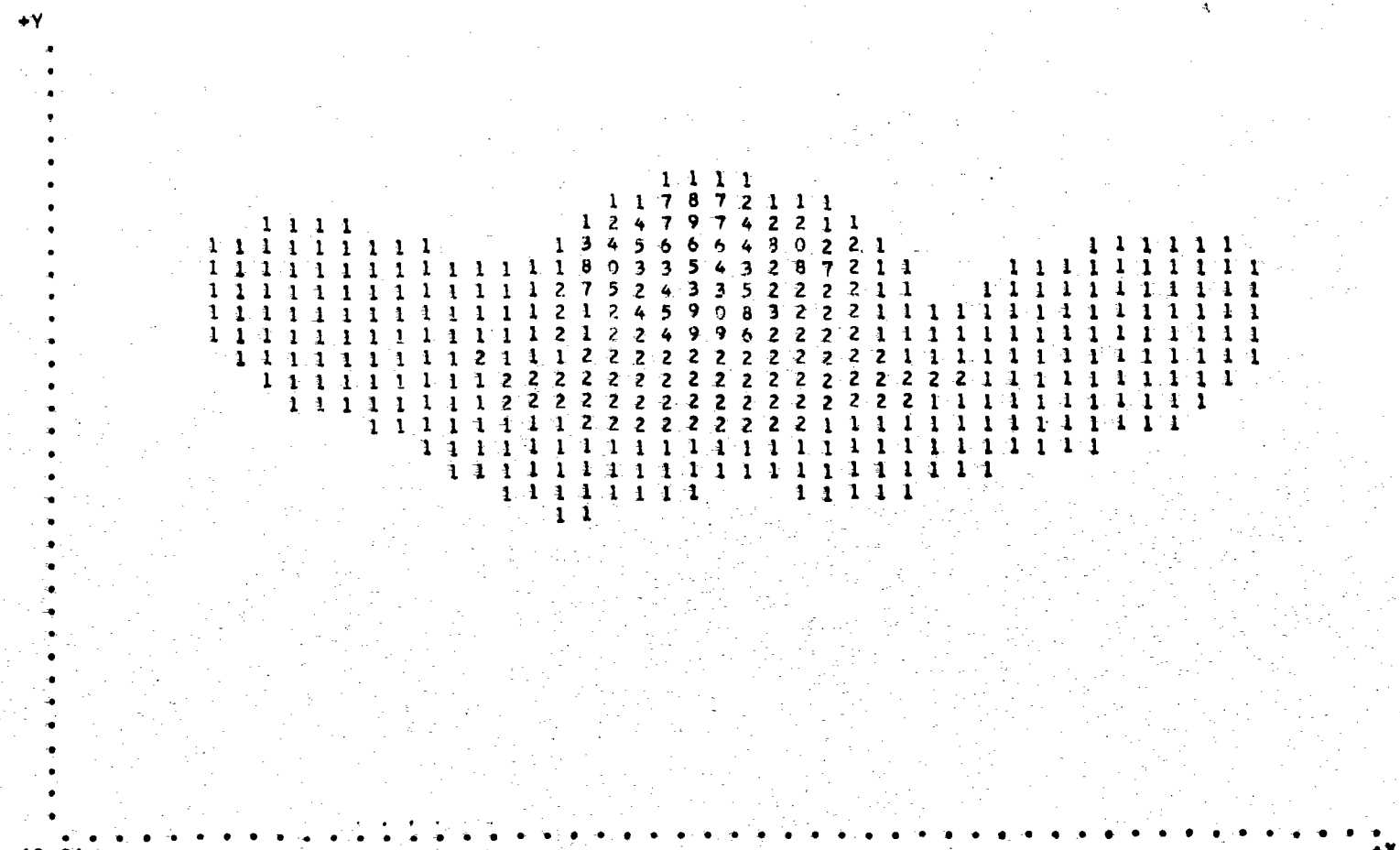
CROSS SECTION 61
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



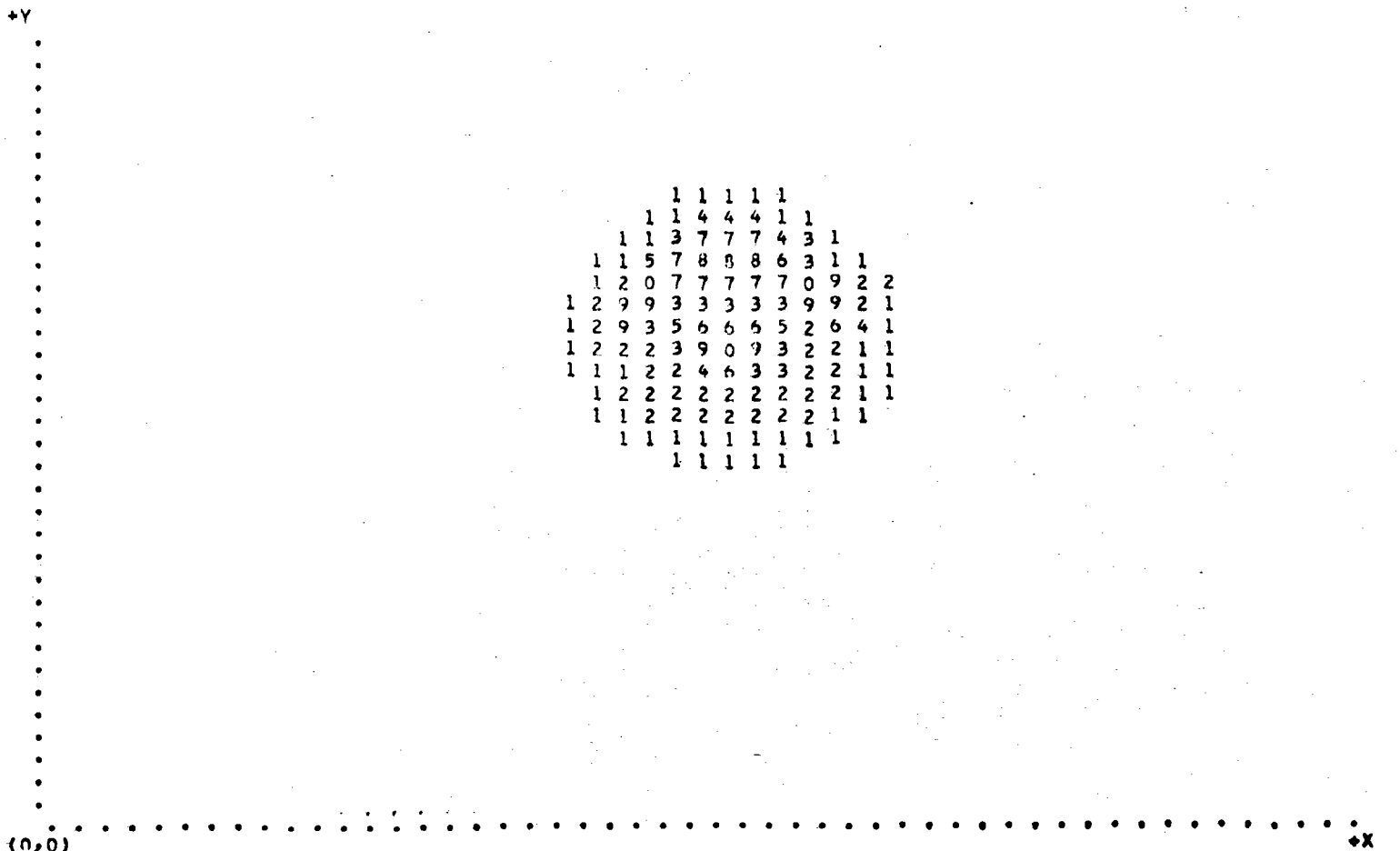
CROSS SECTION 62
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



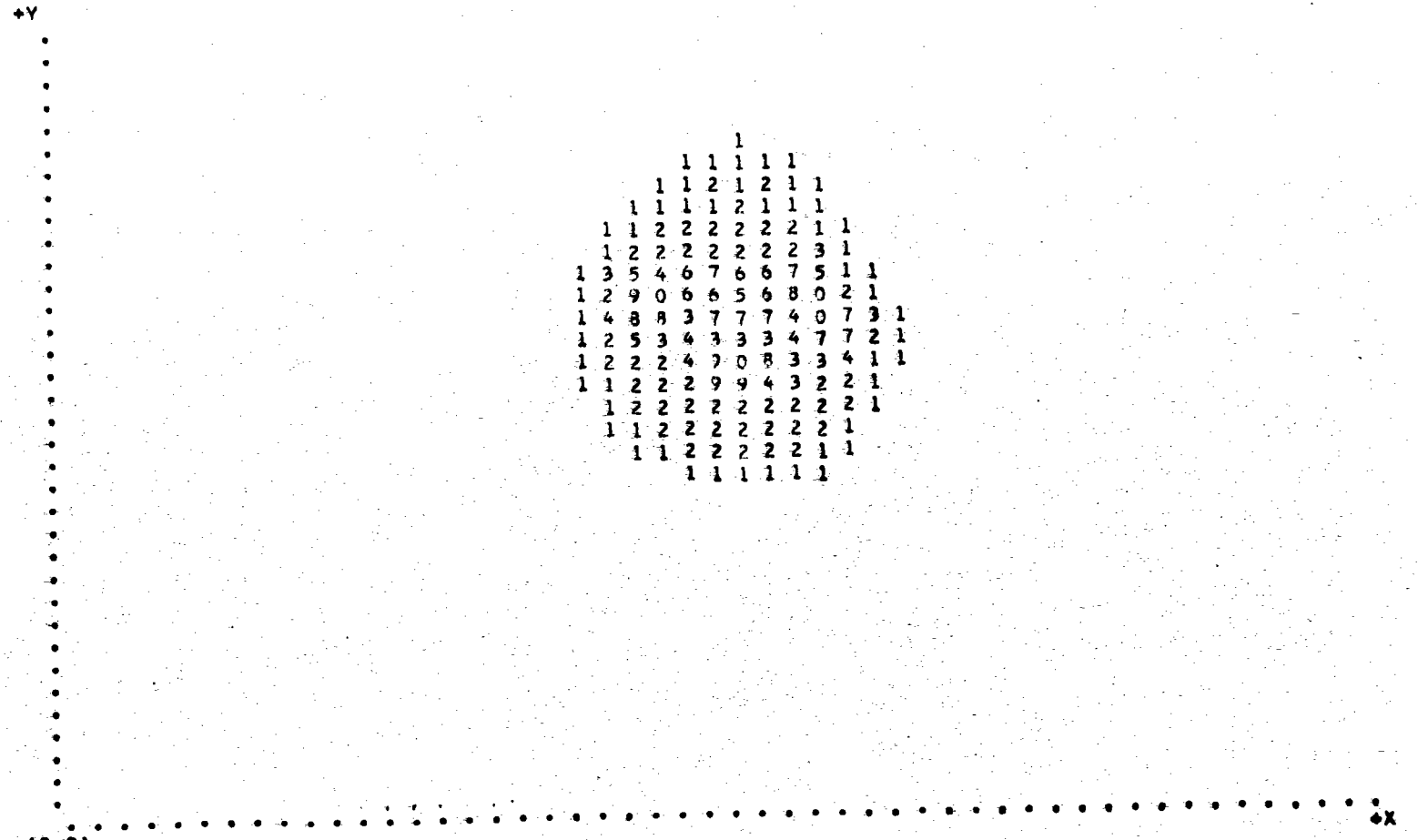
CROSS SECTION 63
AVERAGE DOCTURS EVALUATION 1 HOUR ASSESSMENT



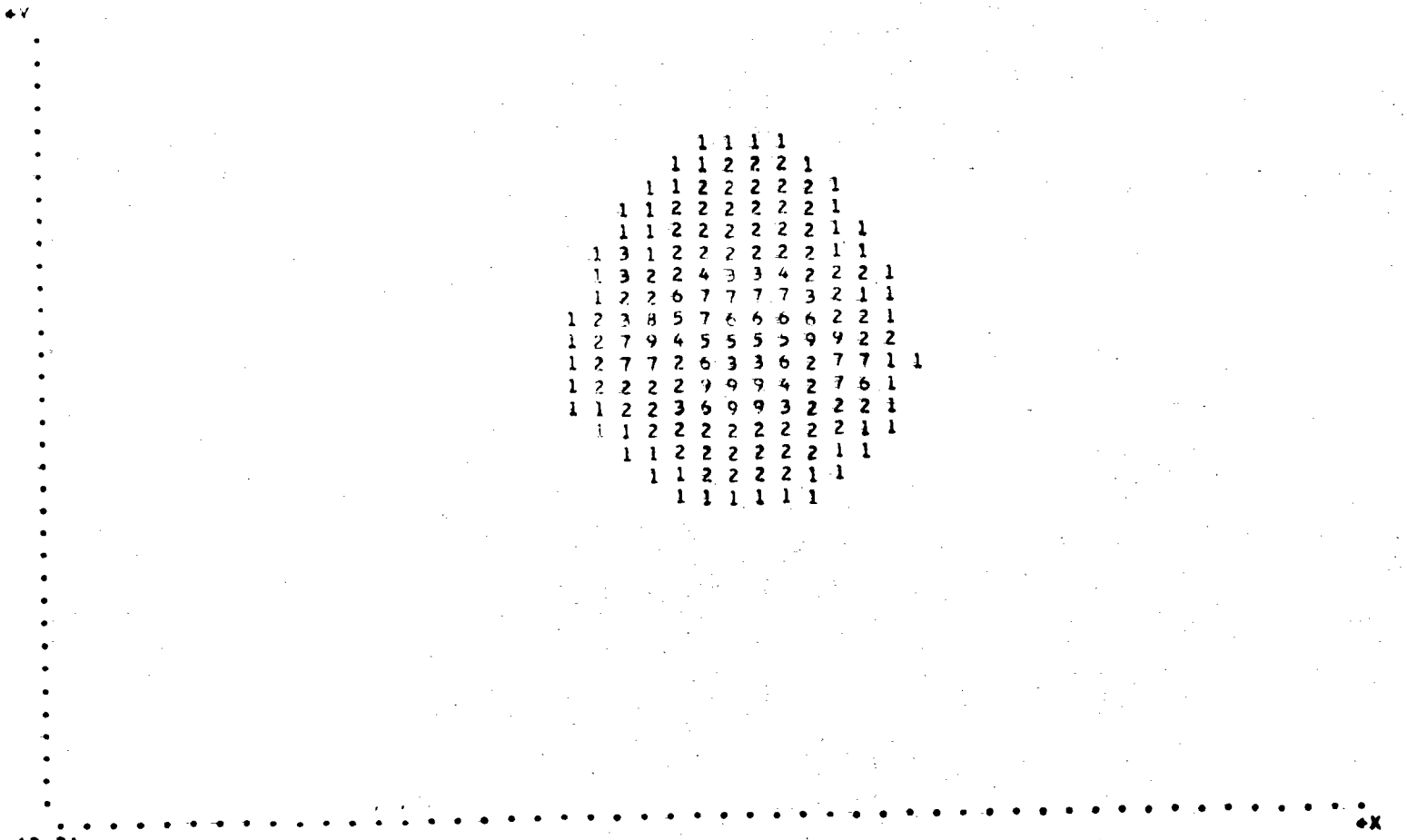
CROSS SECTION 64
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



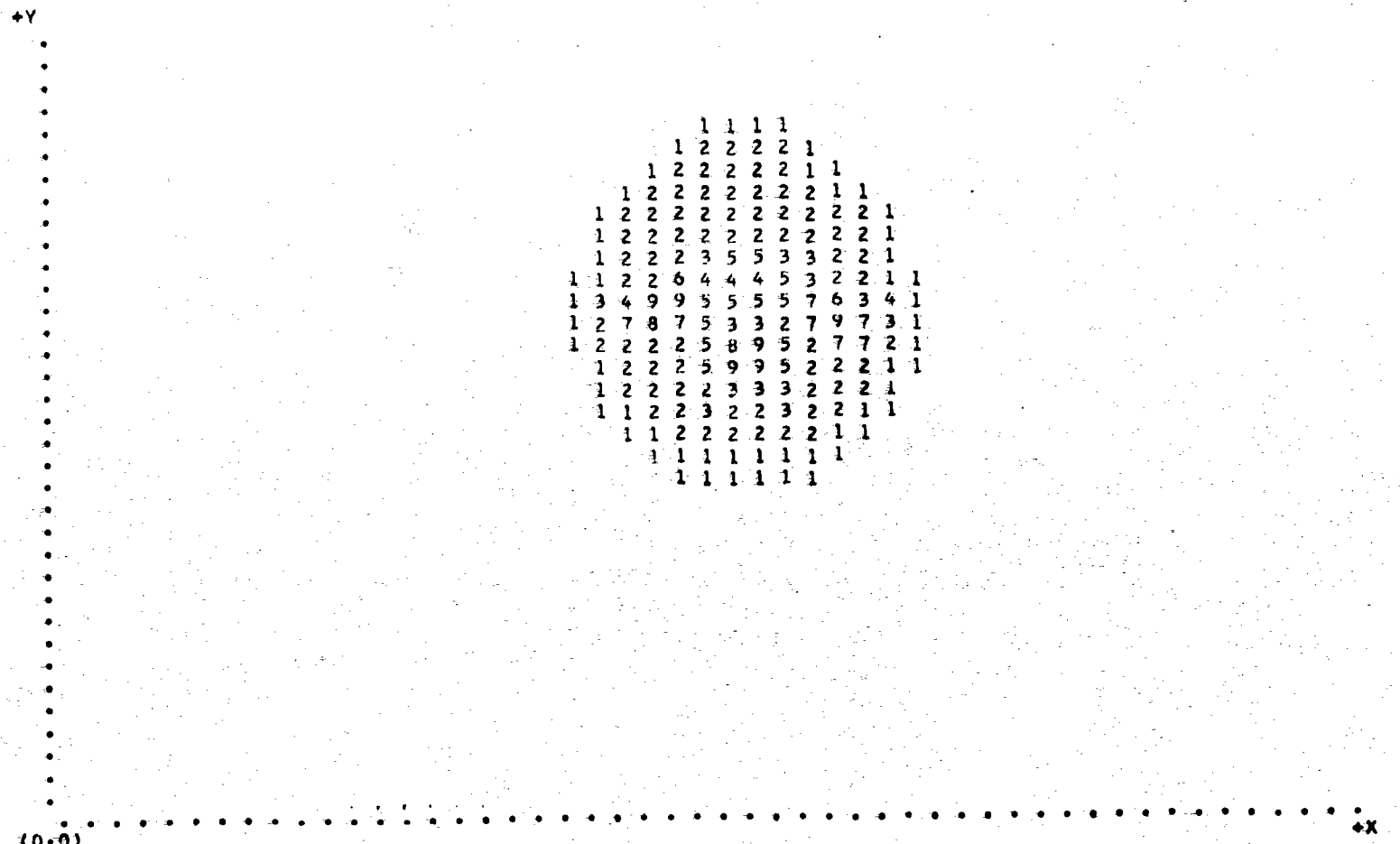
CROSS SECTION 65
AVERAGE DOCTURS EVALUATION 1 HOUR ASSESSMENT



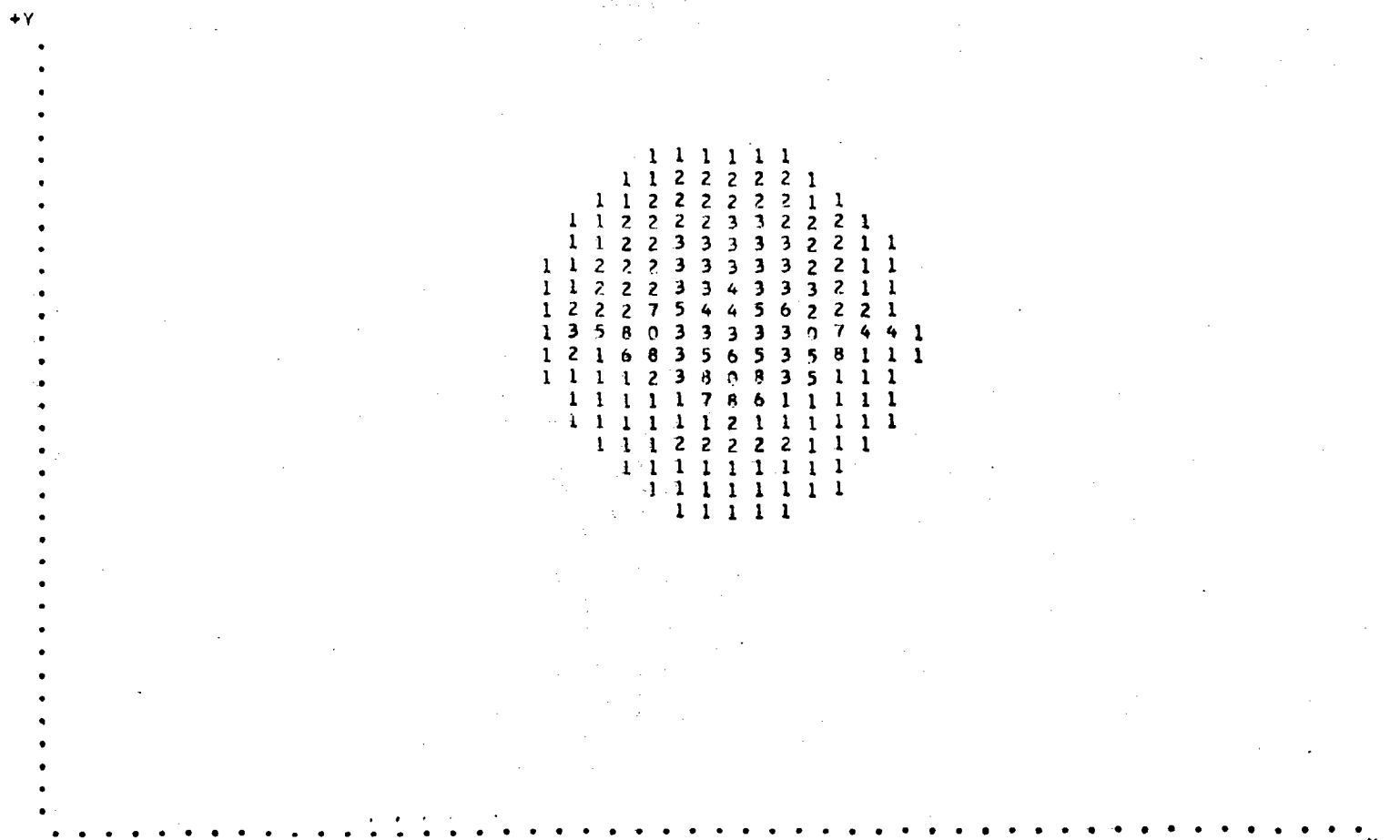
CROSS SECTION 66
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



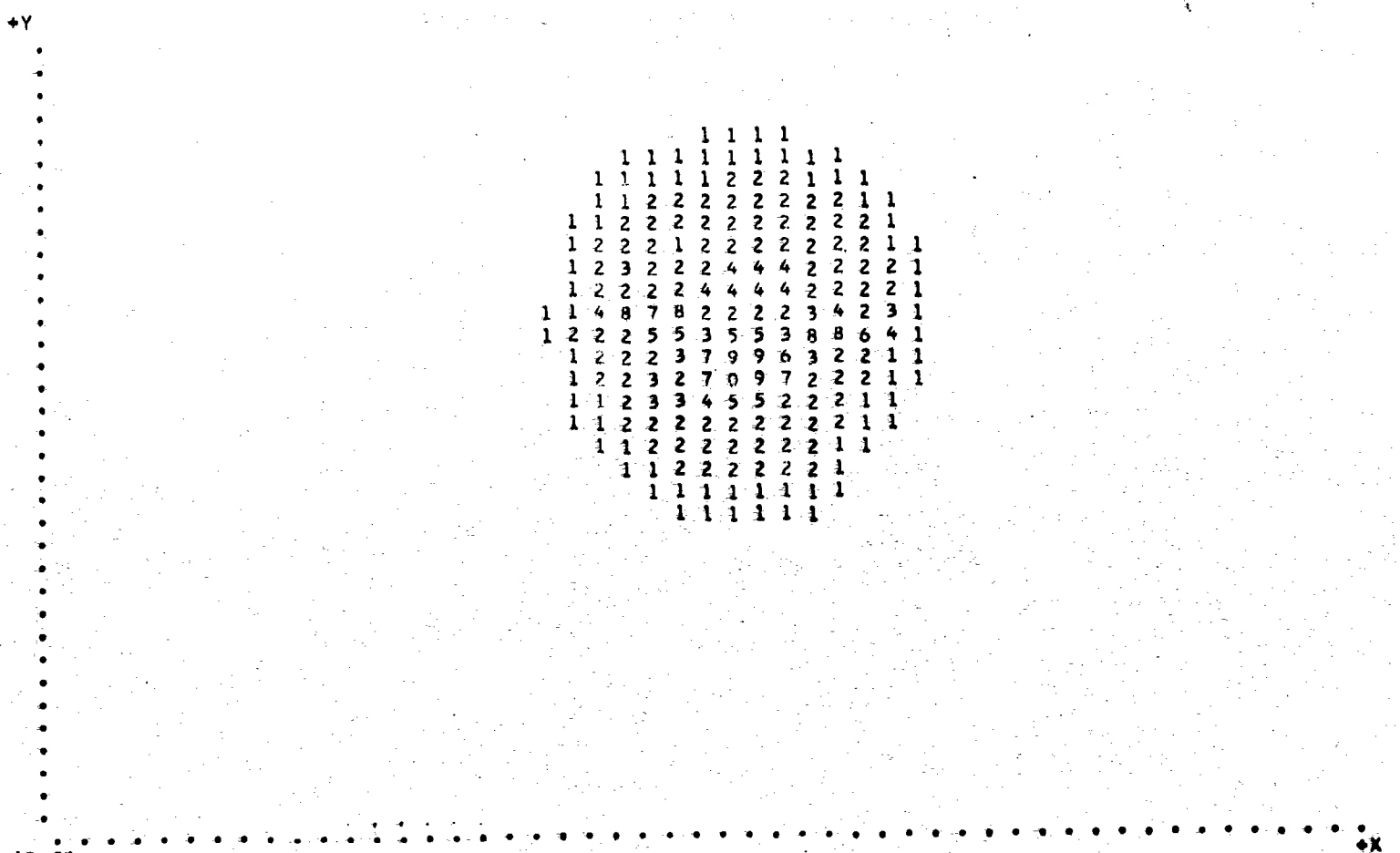
CROSS SECTION 67
AVERAGE DOCTURS EVALUATION 1 HOUR ASSESSMENT



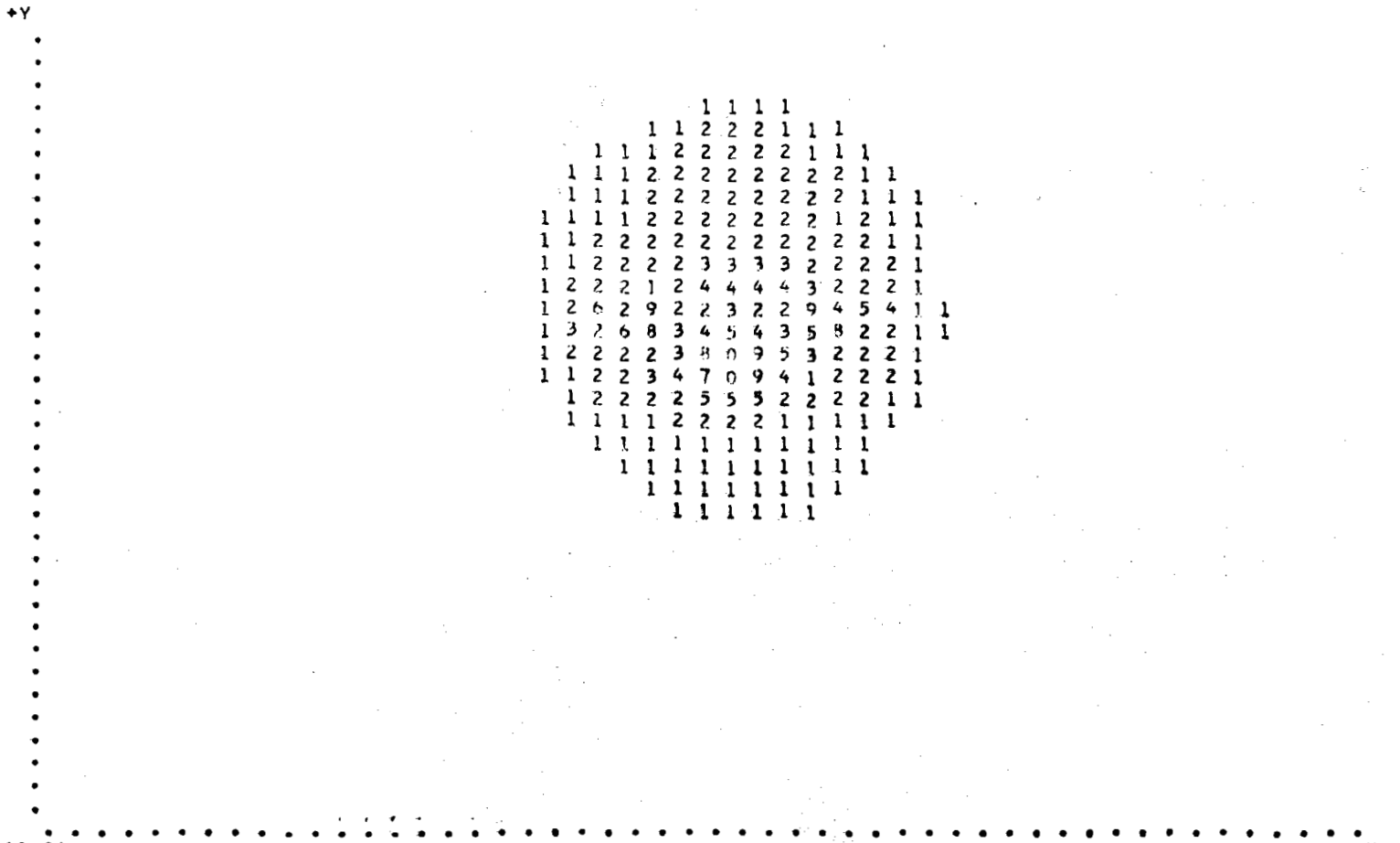
CROSS SECTION 68
AVERAGE DOCTURS EVALUATION 1 HOUR ASSESSMENT



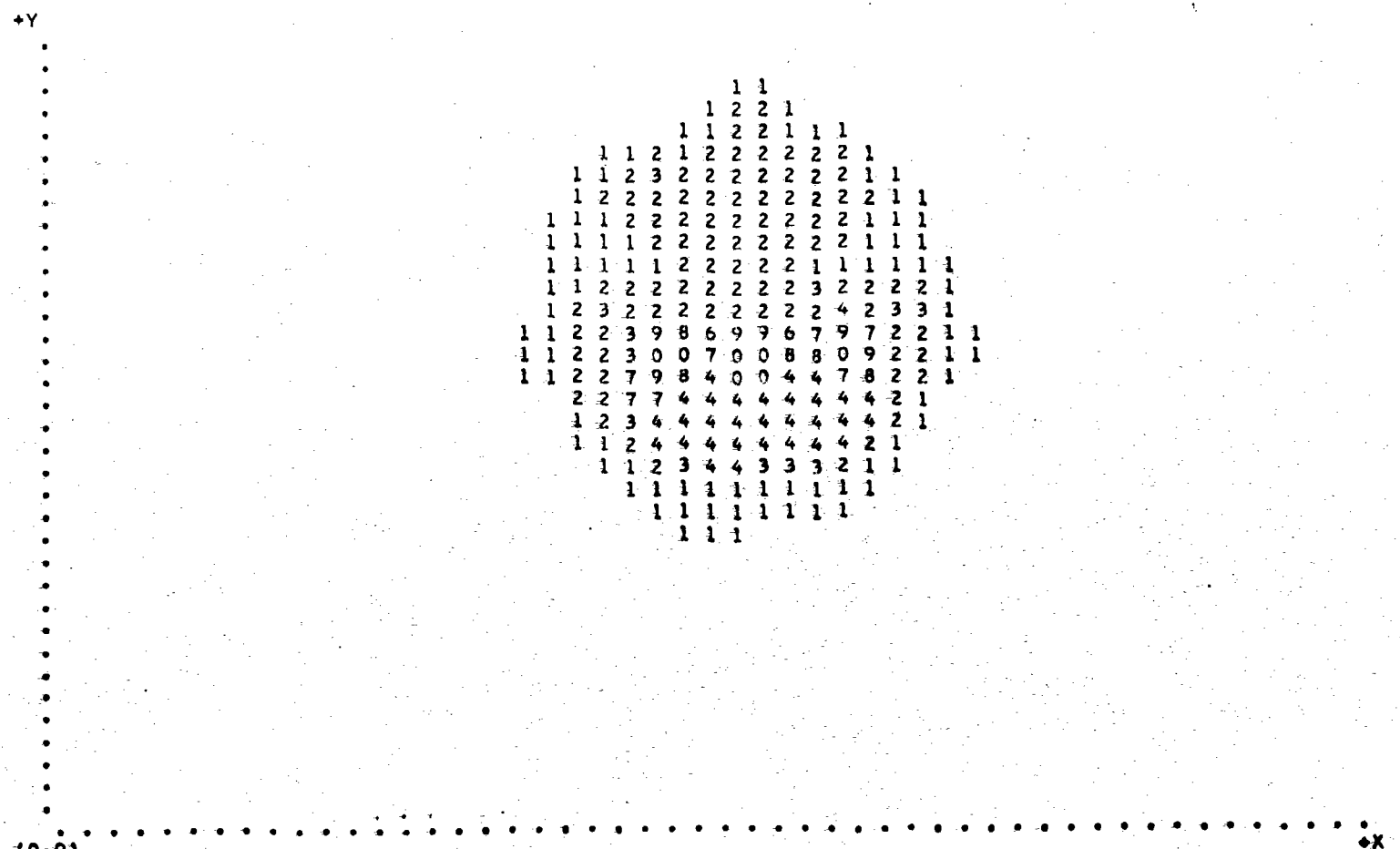
CROSS SECTION 69
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



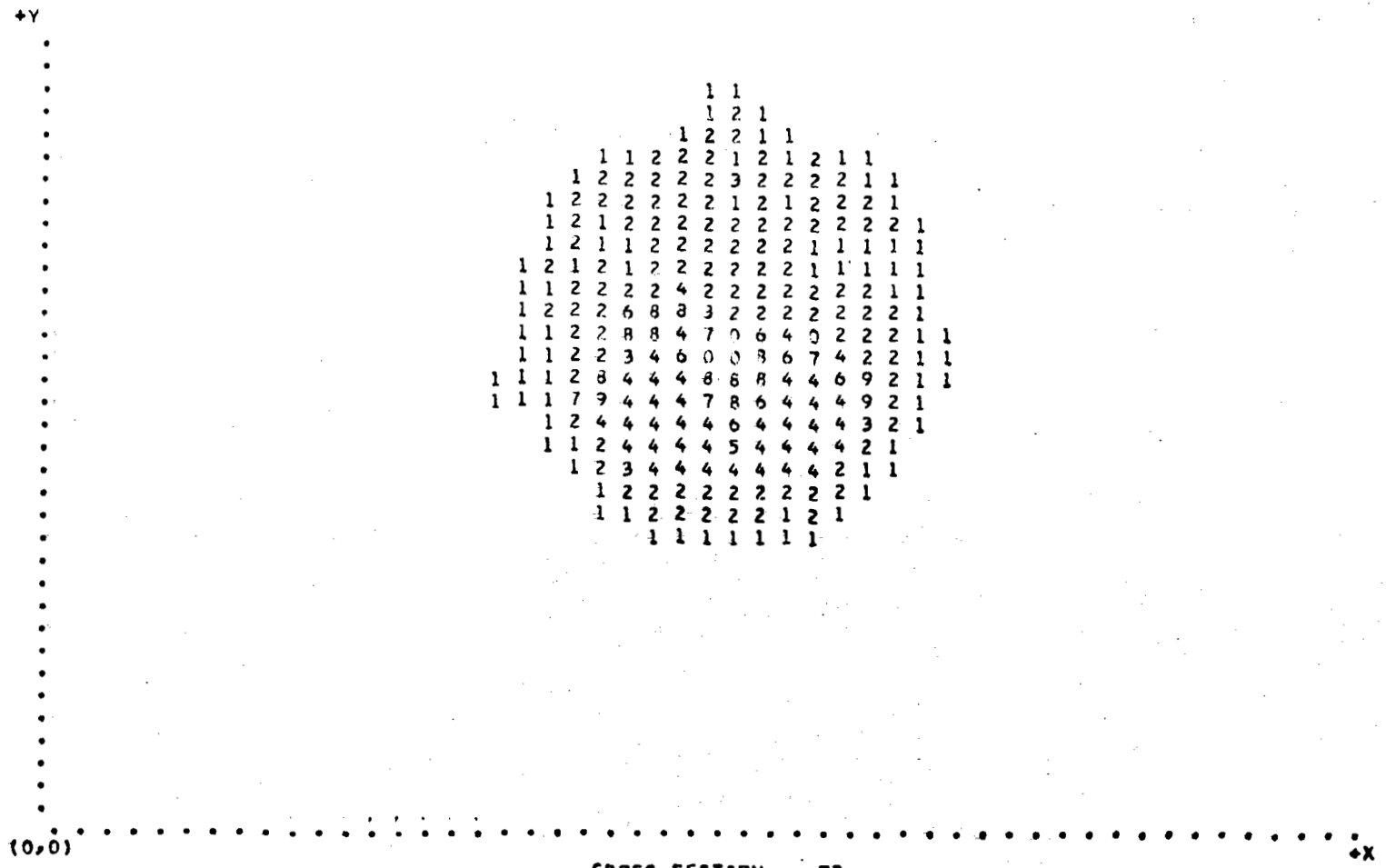
CROSS SECTION 70
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



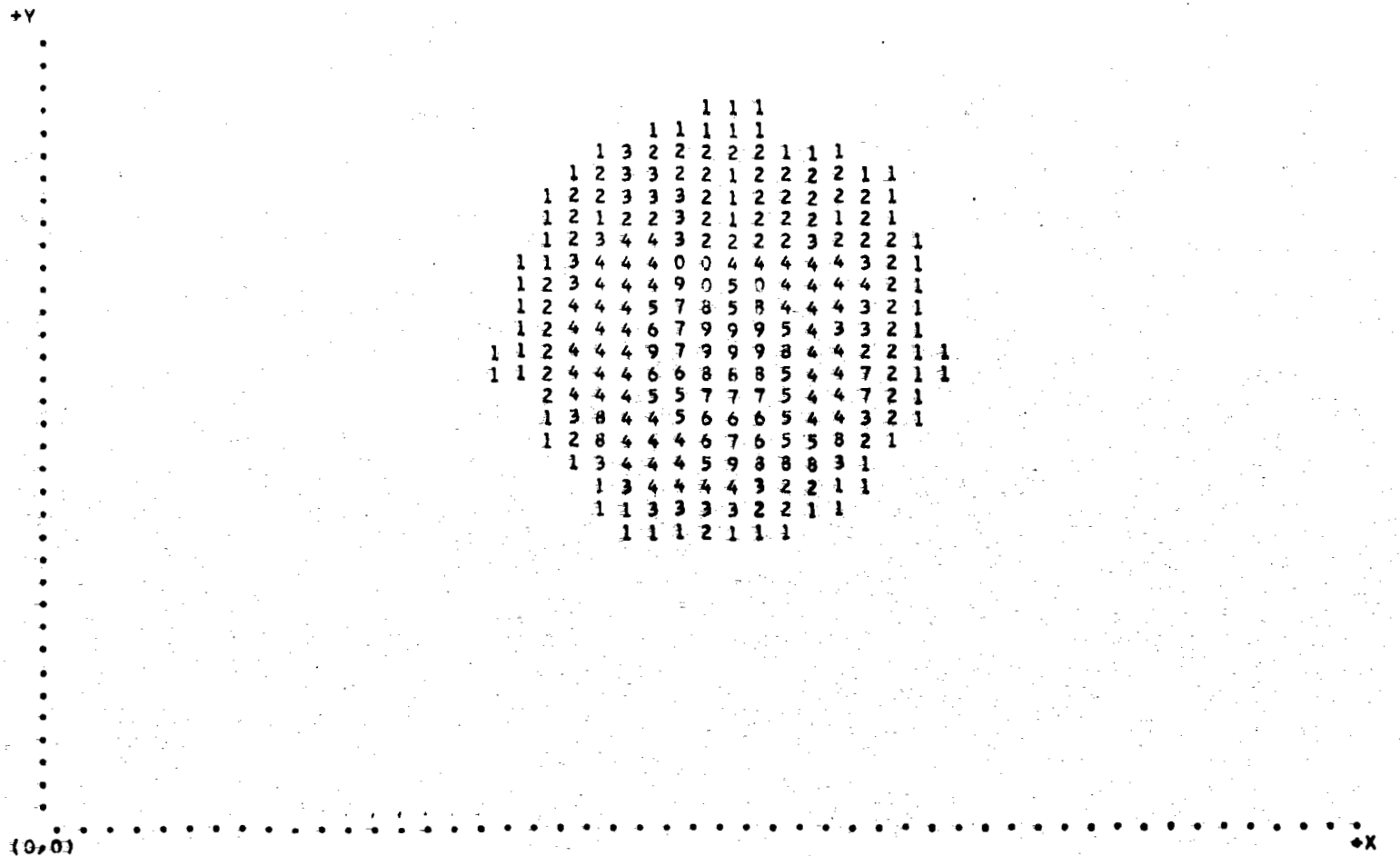
CROSS SECTION 71
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



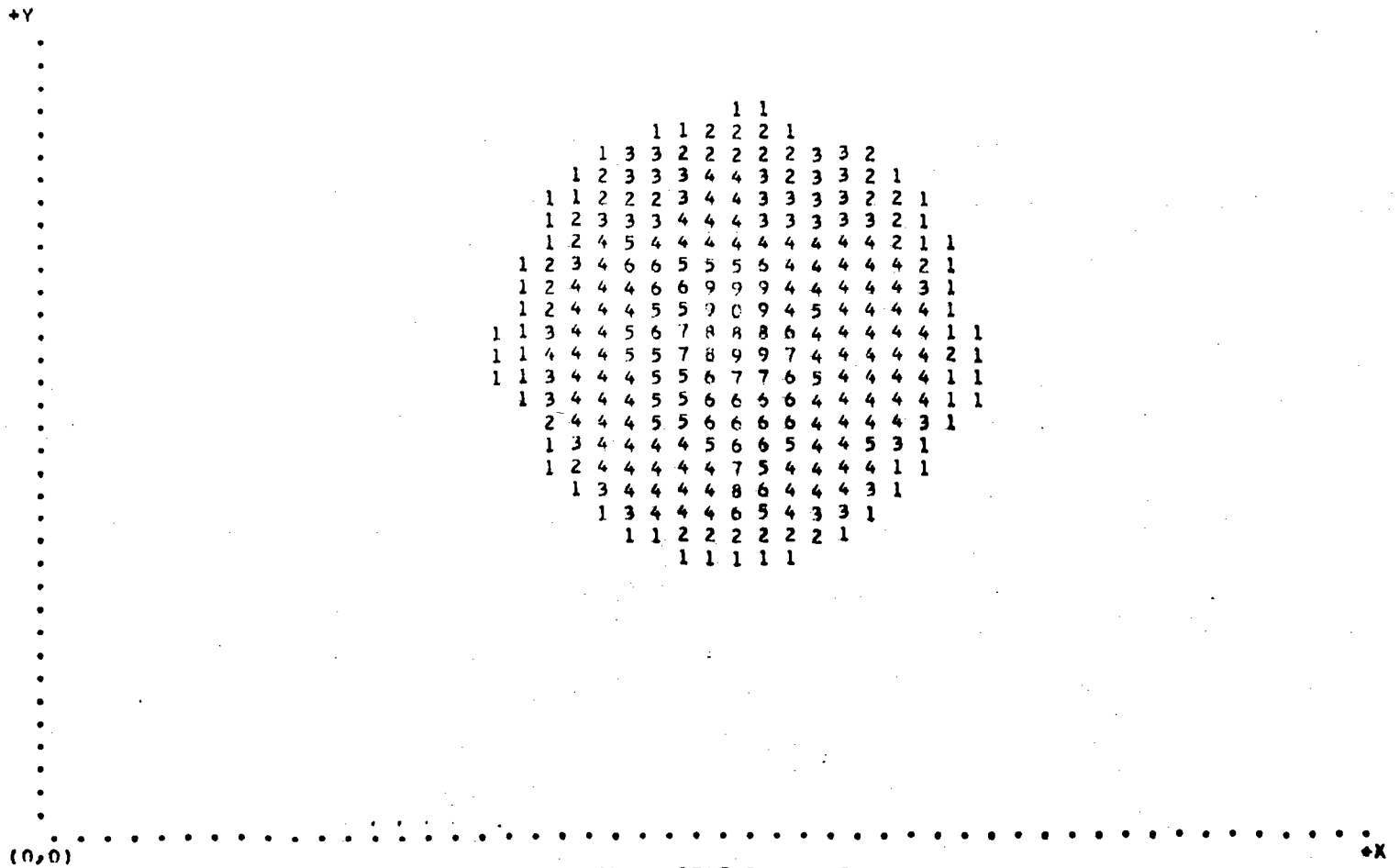
CROSS SECTION 72
AVERAGE DOCTURS EVALUATION 1 HOUR ASSESSMENT



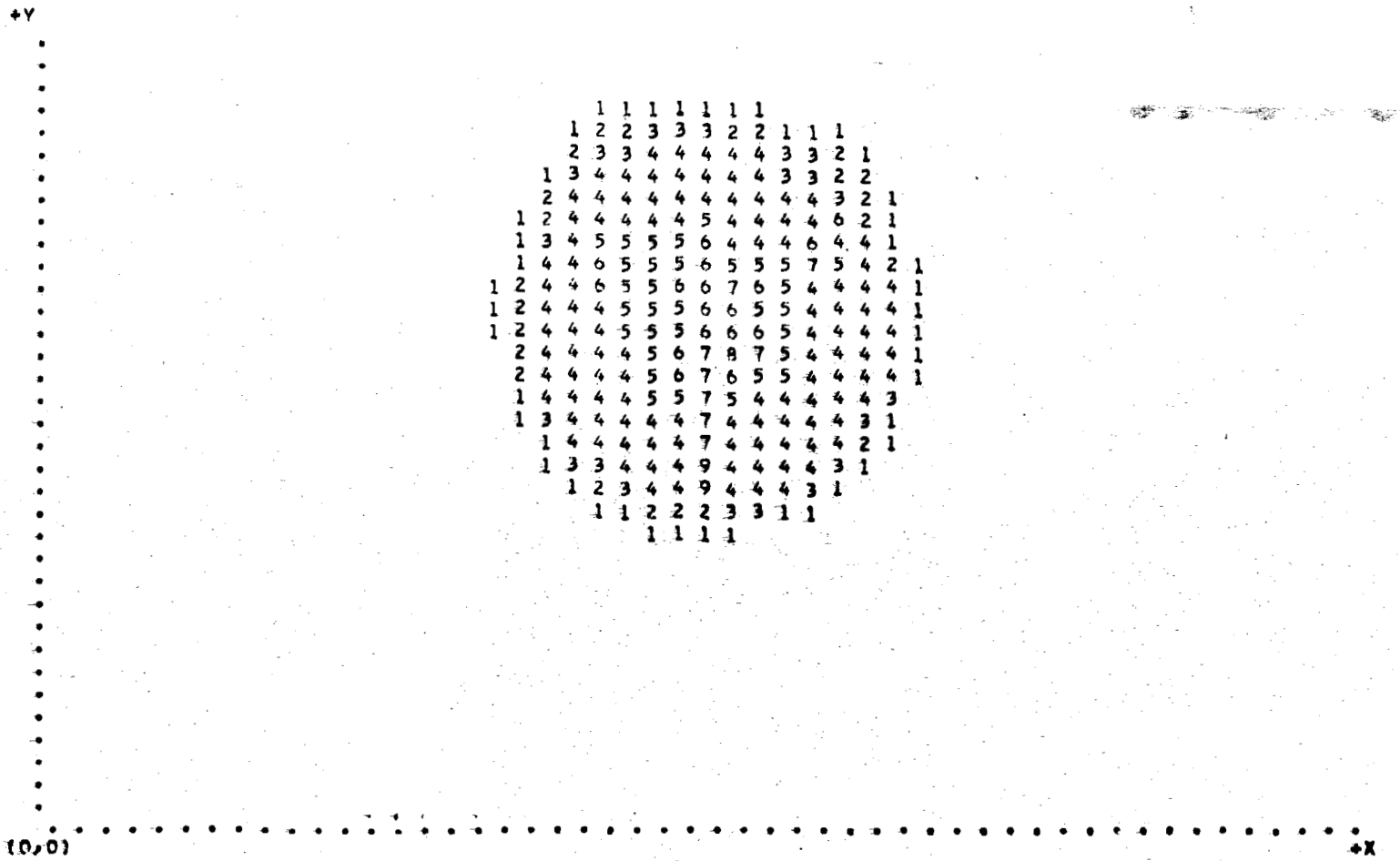
CROSS SECTION 73
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



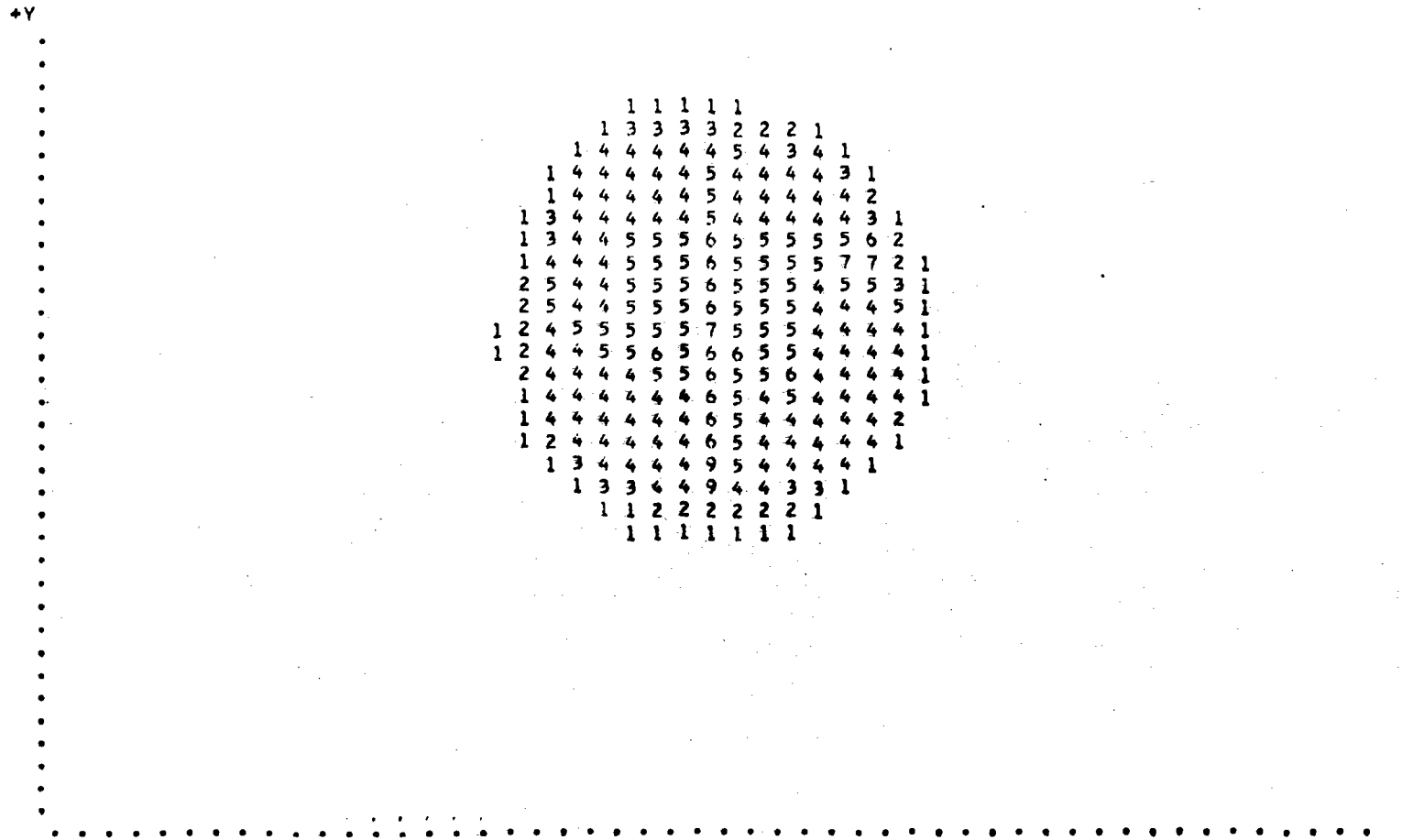
CROSS SECTION 74
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



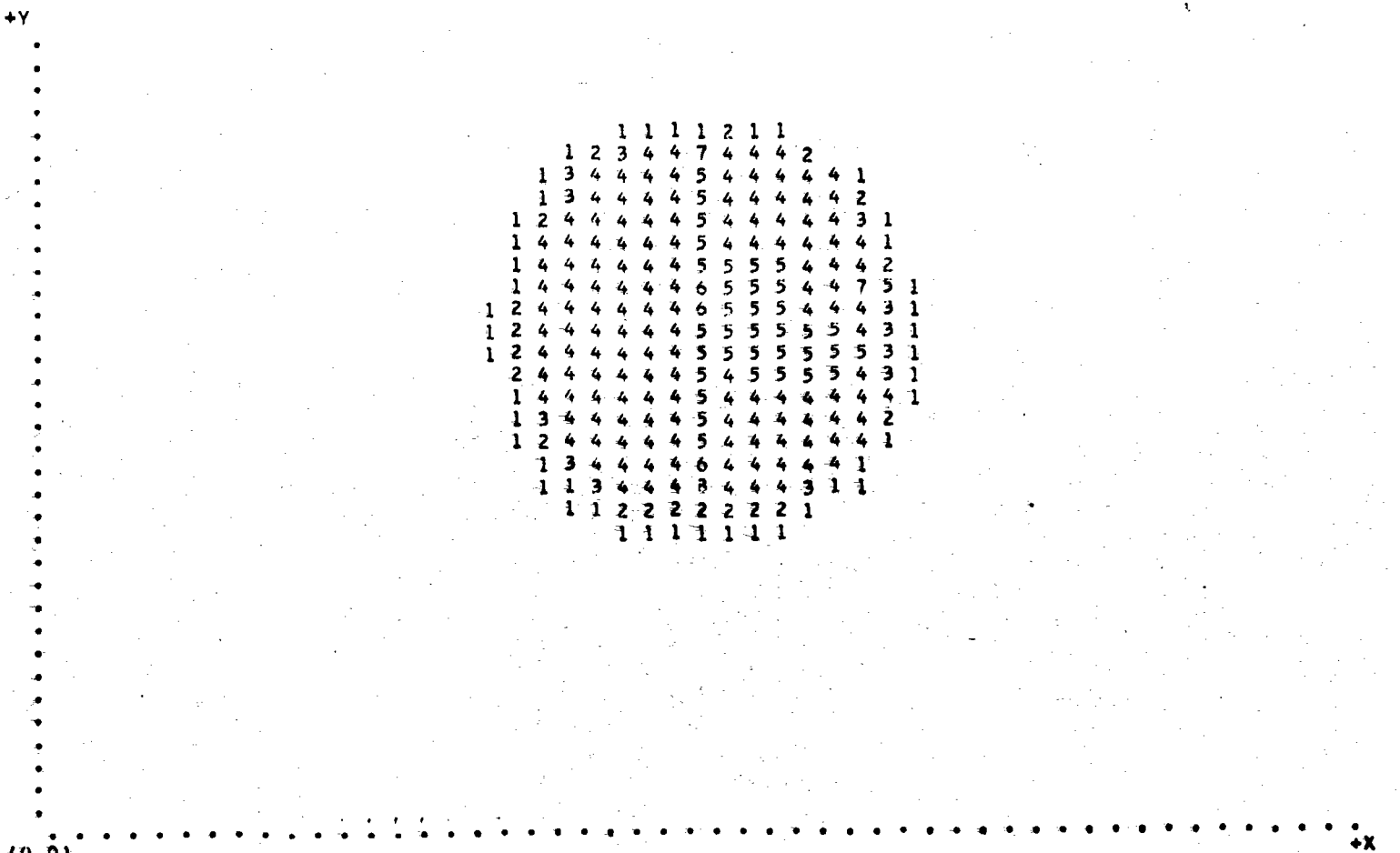
CROSS SECTION 75
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



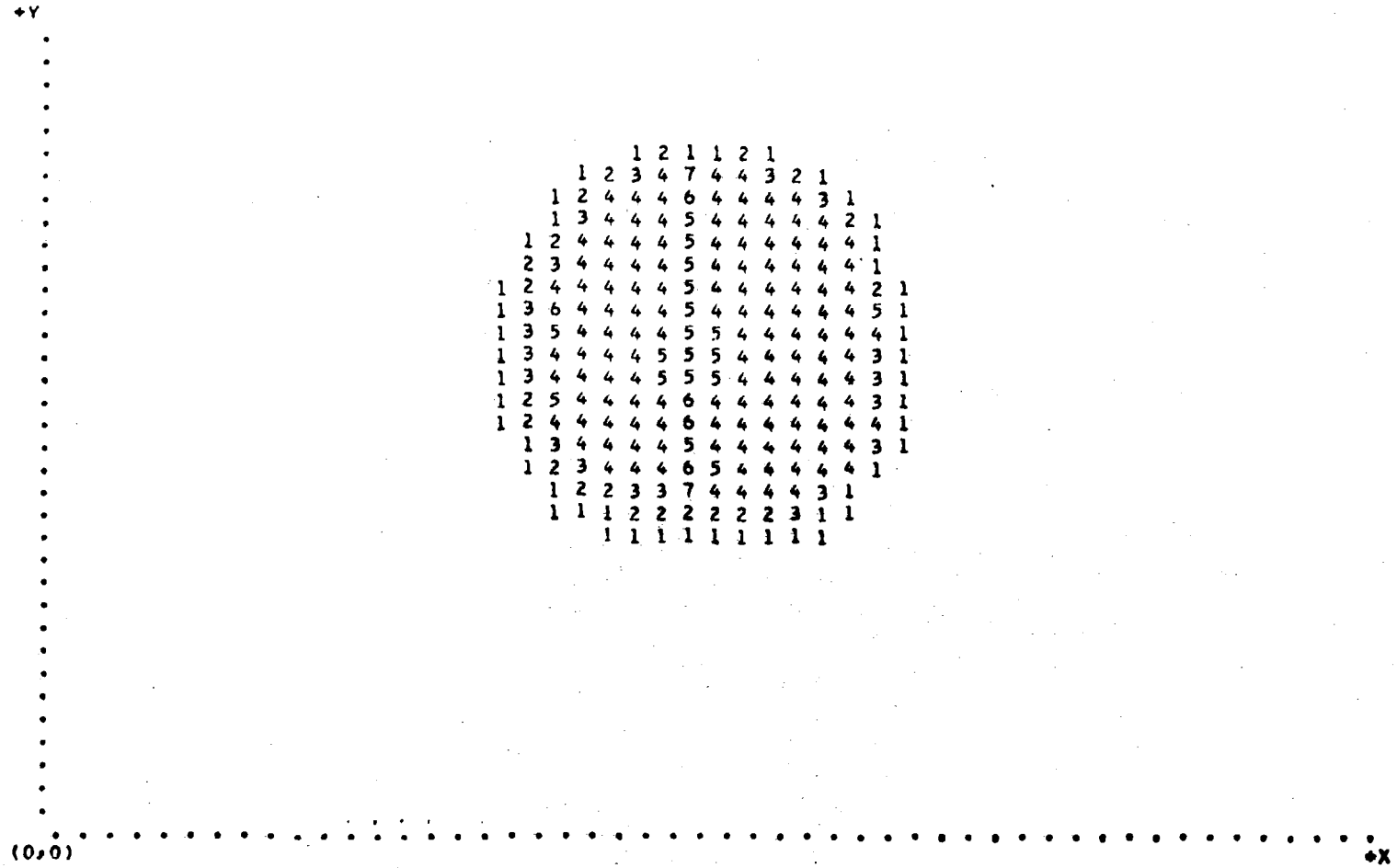
CROSS SECTION 76
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



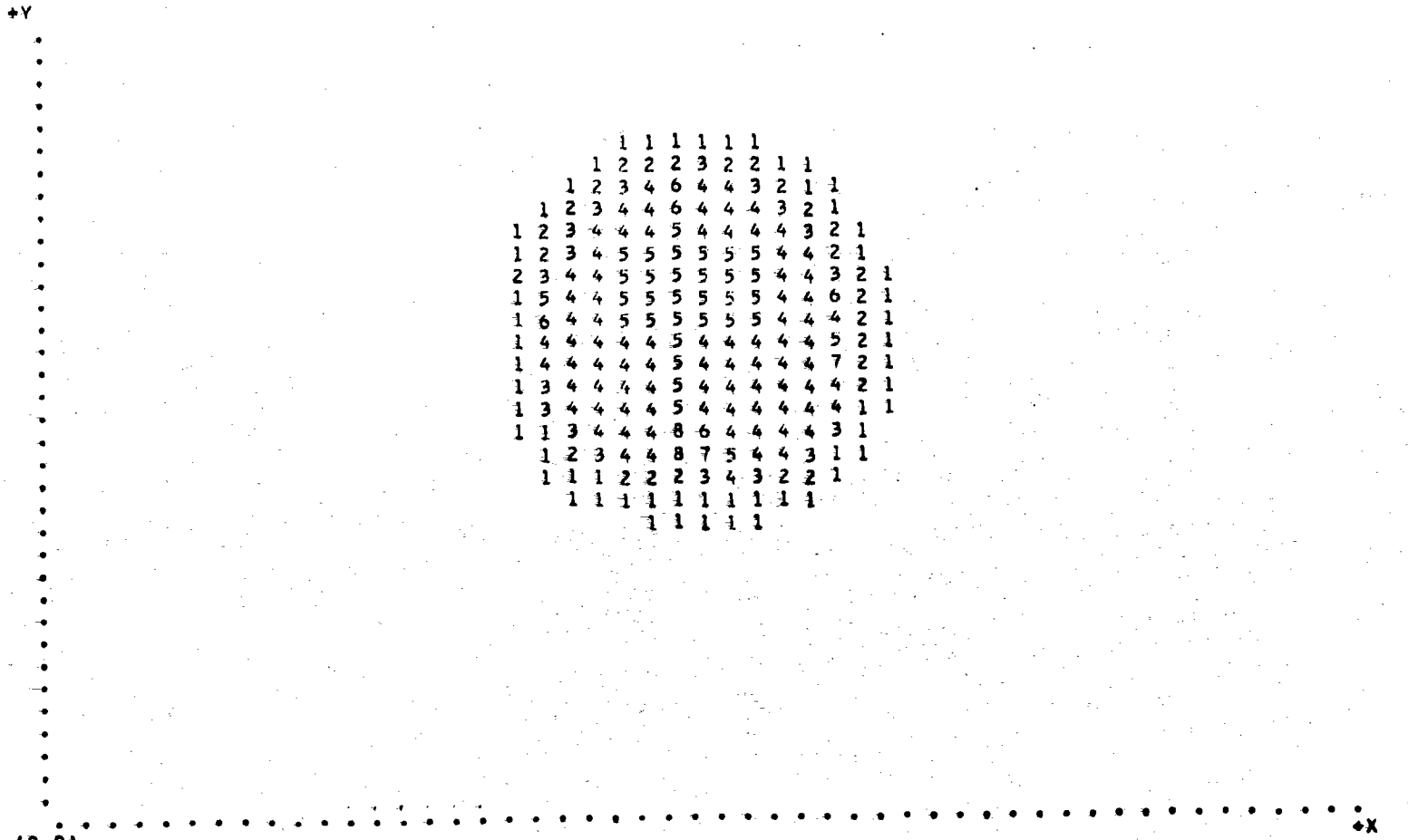
CROSS SECTION 77
AVERAGE DOCTURS EVALUATION 1 HOUR ASSESSMENT



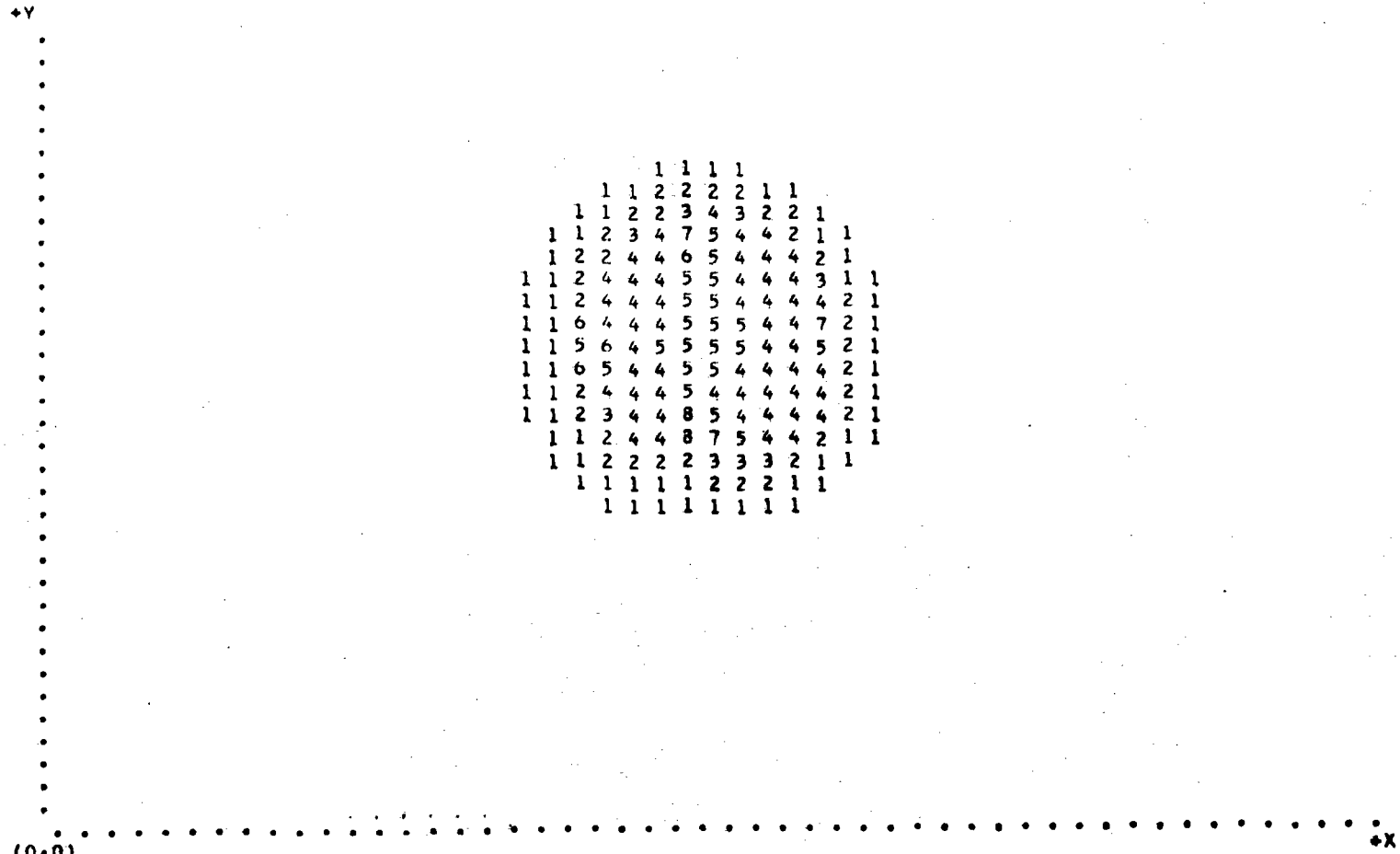
CROSS SECTION 78
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



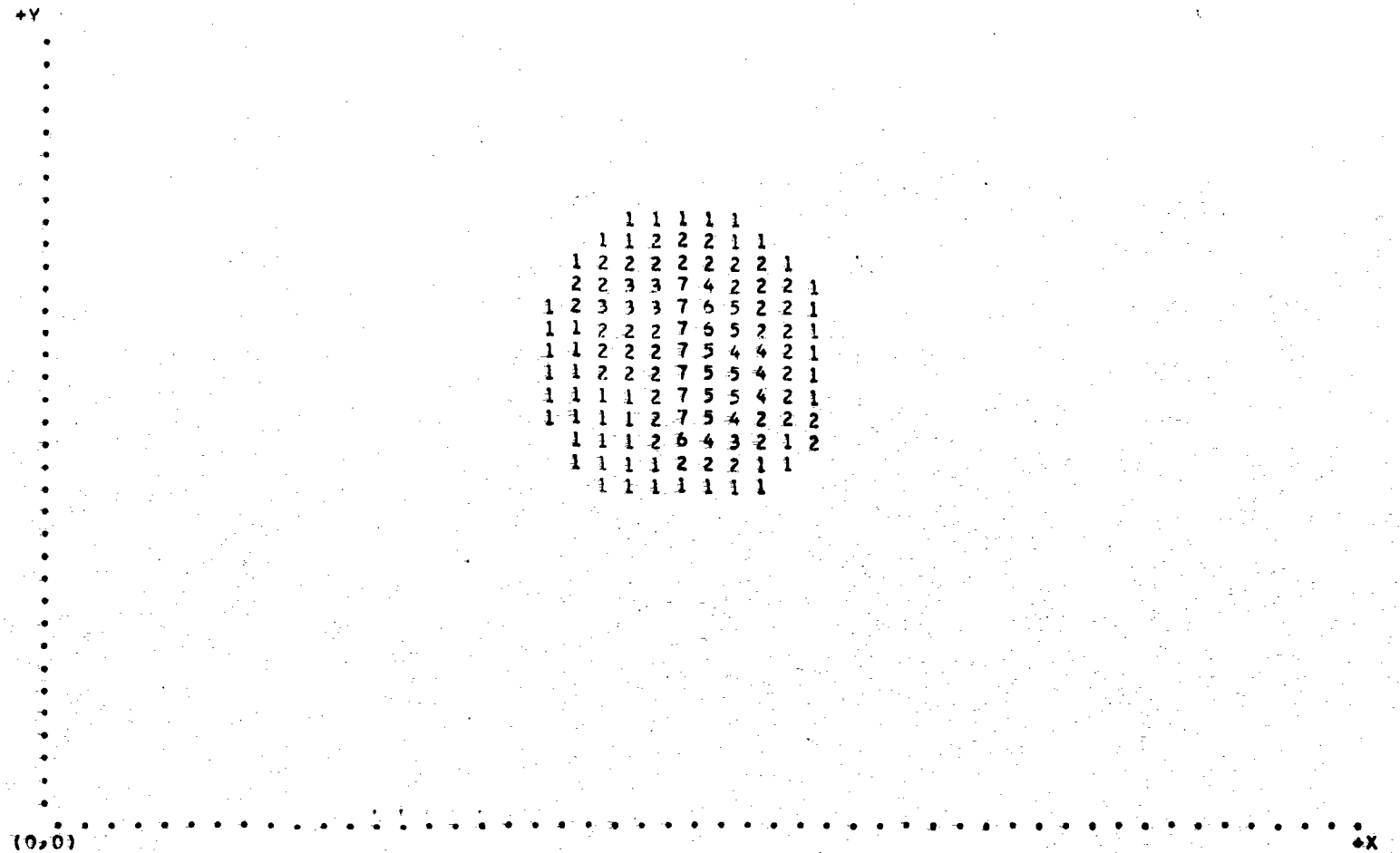
CROSS SECTION 79
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



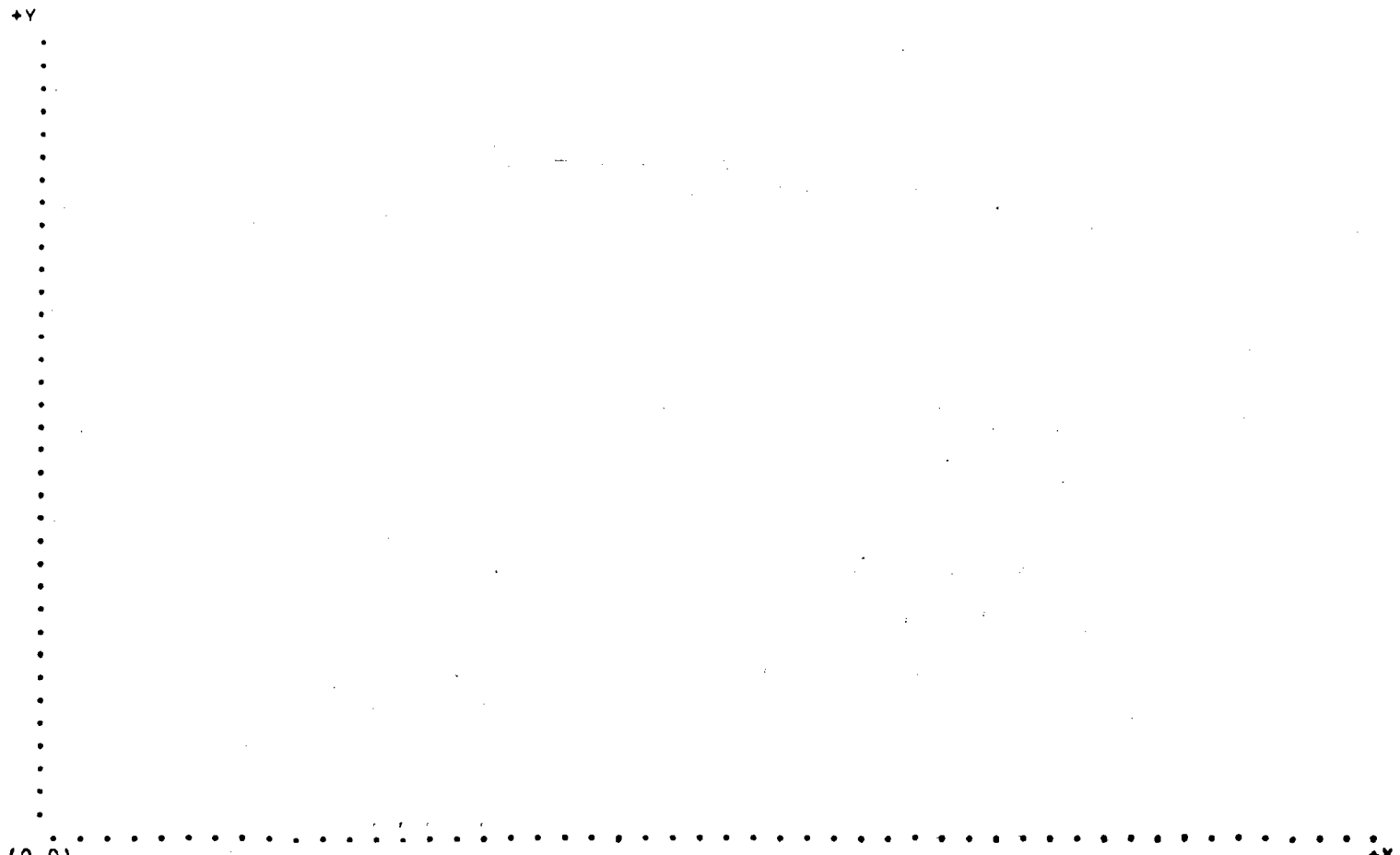
CROSS SECTION 80
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



CROSS SECTION 81
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT

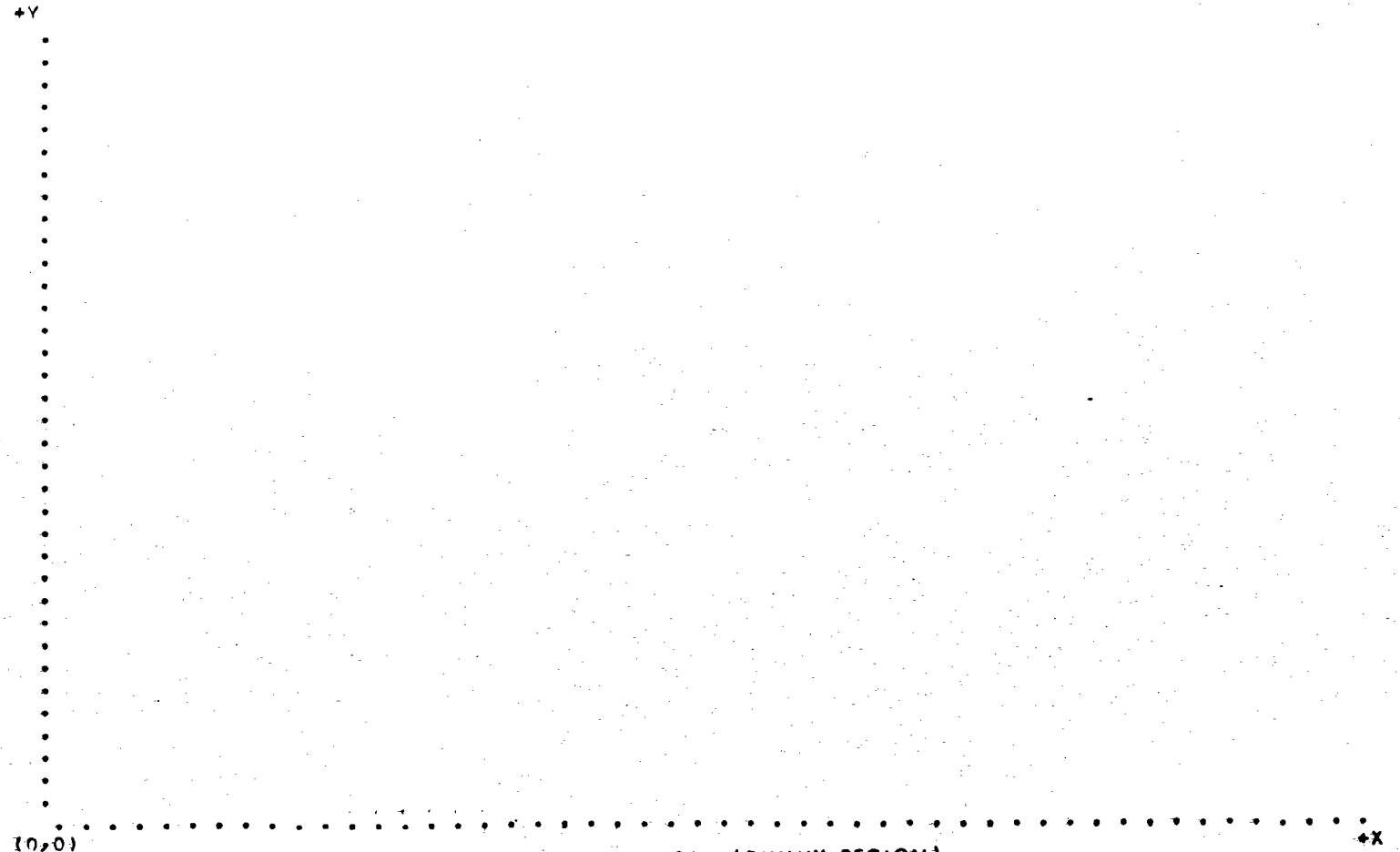


CROSS SECTION 82
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT



CROSS SECTION 83 (DUMMY REGION)

AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT

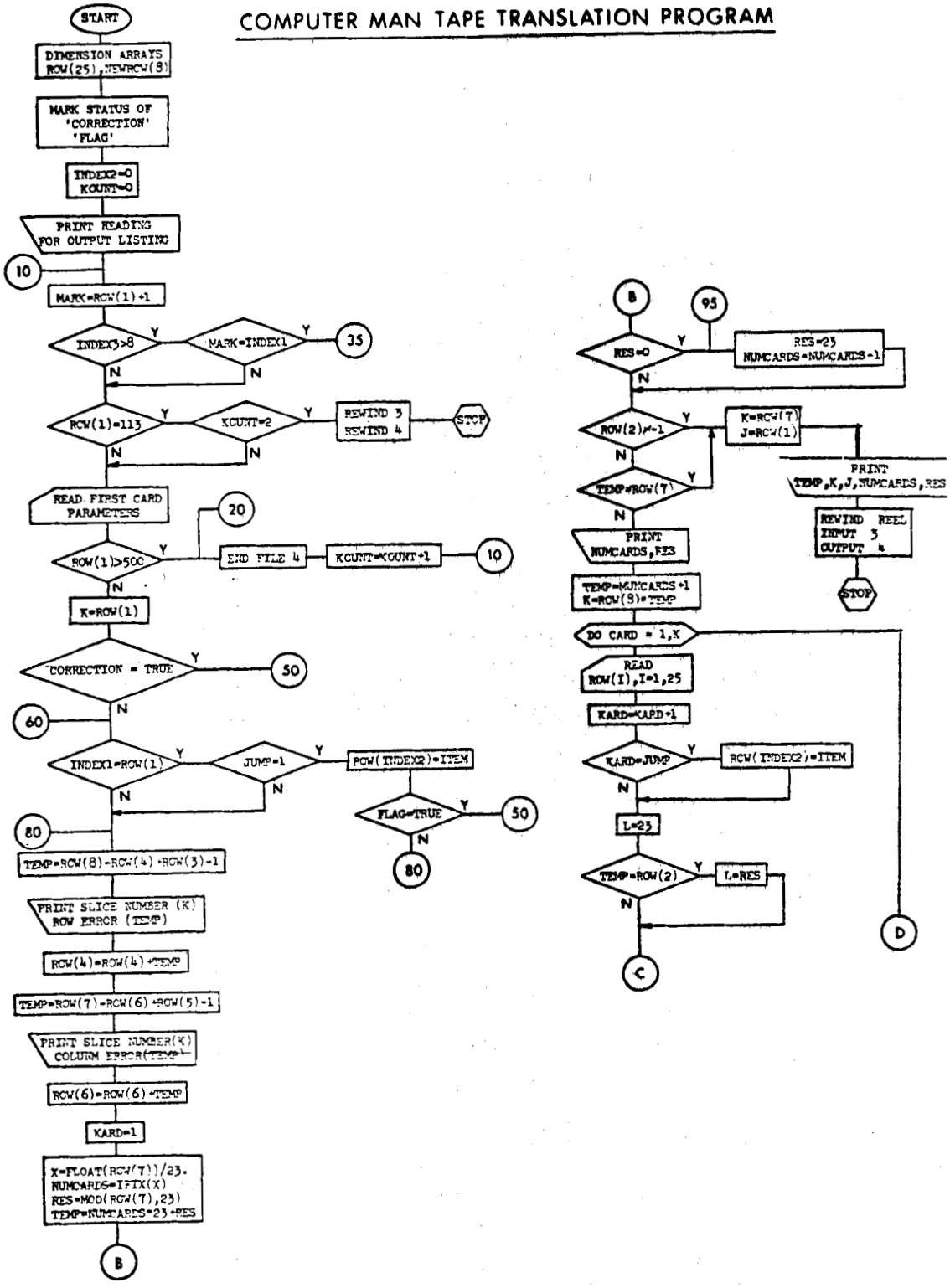


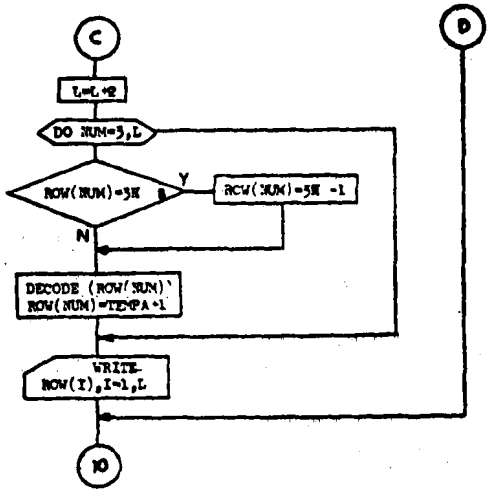
CROSS SECTION 84 (DUMMY REGION)
AVERAGE DOCTORS EVALUATION 1 HOUR ASSESSMENT

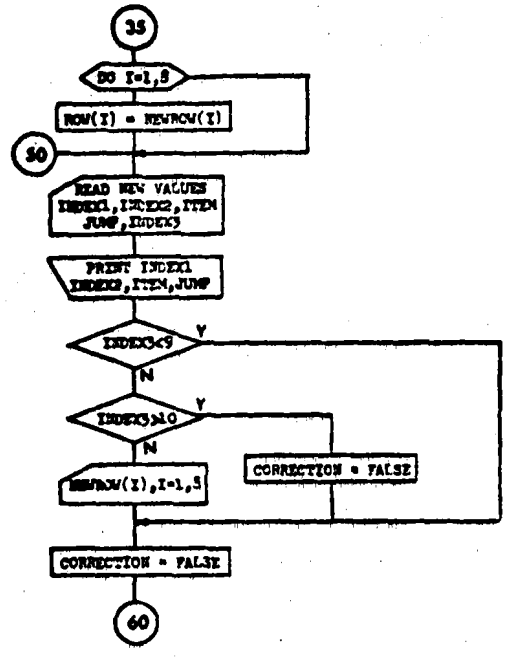
APPENDIX D

FLOW CHARTS AND PROGRAM LISTINGS ASSOCIATED WITH THE COMPUTER PROGRAMS CONTAINED IN THIS REPORT

COMPUTER MAN TAPE TRANSLATION PROGRAM



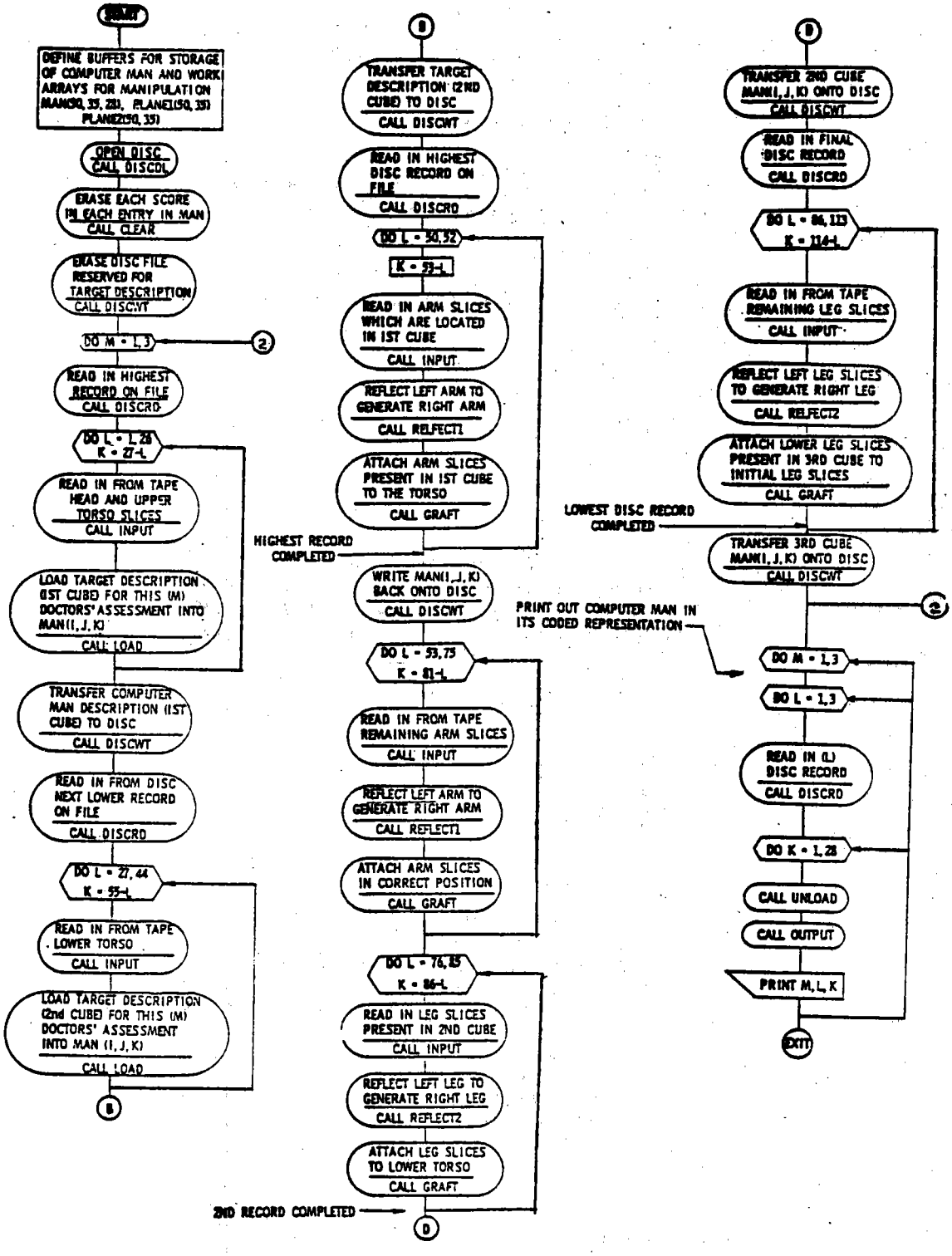

C
L=L+2
DO NUM=3,L
ROW(NUM)=5K
Y
ROW(NUM)=5K -1
N
DECODE (ROW(NUM)'
ROW(NUM)=TEMPA+1
WRITE
ROW(I),I=1,L
10
D

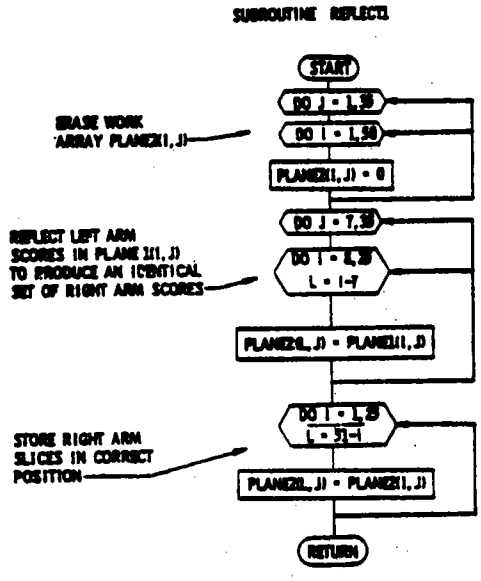

35
DO I=1,9
ROW(I) = NEWROW(I)
50
READ NEW VALUES
INDEX1,INDEX2,ITEM
JUMP,INDEX3
PRINT INDEX1
INDEX2,ITEM,JUMP
INDEX3<9
Y
N
INDEX3>10
Y
N
NEWROW(I),I=1,9
CORRECTION = FALSE
CORRECTION = FALSE
60

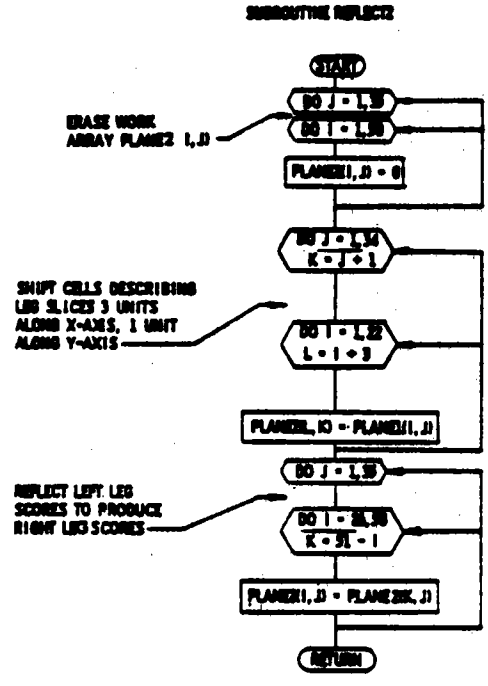
```
*       LIST(START)
$       COMM REEL 24A11 TO UNIT 3 INPUT
$       COMM REEL 26C14 TO UNIT 4 OUTPUT
$       MAXO(15000)LINES
$       MAXT(10)MINS
        DIMENSION ROW(25),NEWROW(8)
        INTEGER ROW,TEMP,CARD,RES,TEMPA
        LOGICAL CORRECTION,FLAG
C
C        CORRECTION = THERE ARE CORRECTIONS TO BE MADE
C        FLAG = APPLY CORRECTION TO THIS LINE
C        MARK FLAG WHERE SLICE IS TO BE CORRECTED
C
        CORRECTION=.TRUE.
        FLAG=.TRUE.
        INDEX2=0
        KOUNT=0
        WRITE(6,111)
111     FORMAT('OROW AND COLUMN COUNTS ARE A CHECK ON THE PARAMETERS',
       A        ' THAT POSITION THE SUBMATRIX WITHIN THE SLICE.',
       B        /,'0THIS PROGRAM EDITS THE FIRST CARD S PARAMETER.')
10      MARK=ROW(1)+1
C
C        INDEX3  KEY
C        8    CHANGE CARD ENTRY
C        9   READ IN NEW HEAD CARD
C         10   APPLY NO MORE CORRECTIONS
C
        IF(INDEX3.GT.8.AND.MARK.EQ.INDEX1)GO TO 35
        IF(ROW(1).EQ.113.AND.KOUNT.EQ.2) GO TO 30
        READ(3,108)(ROW(I),I=1,8)
108     FORMAT(I3,I2,4I4,4X,2I4)
        IF(ROW(1).GT.500) GO TO 20
        K=ROW(1)
        IF(CORRECTION)GOTO 50
60      IF(INDEX1.EQ.ROW(1) .AND. JUMP.EQ.1) GOTO 70
80      TEMP=ROW(8)-ROW(4)+ROW(3)-1
        WRITE(6,103)K,TEMP
103     FORMAT('0SLICE',2X,I5,2X,'ROW COUNT OFF BY',2X,I5)
        ROW(4)=ROW(4)+TEMP
        TEMP=ROW(7)-ROW(6)+ROW(5)-1
        WRITE(6,104)K,TEMP
104     FORMAT('0SLICE',2X,I5,2X,'COLUMN COUNT OFF BY',2X,I5)
        ROW(6)=ROW(6)+TEMP
        KARD=1
        WRITE(4,100)(ROW(I),I=1,8)
100     FORMAT(I3,I2,4I4,4X,2I4)
        X=FLOAT(ROW(7))/23.
        NUMCARDS=IFIX(X)
        RES=MOD(ROW(7),23)
        TEMP=NUMCARDS*23+RES
        IF(RES.EQ.0)GO TO 95
        IF(ROW(2).NE.-1)GO TO 40
90      IF(TEMP.NE.ROW(7))GOTO 40
        WRITE(6,105)NUMCARDS,RES
105     FORMAT('0NUMBER OF EVEN CARDS PER COLUMN',2X,I5,2X,'WITH',
       1        2X,I5,2X,'LEFT OVER')
```

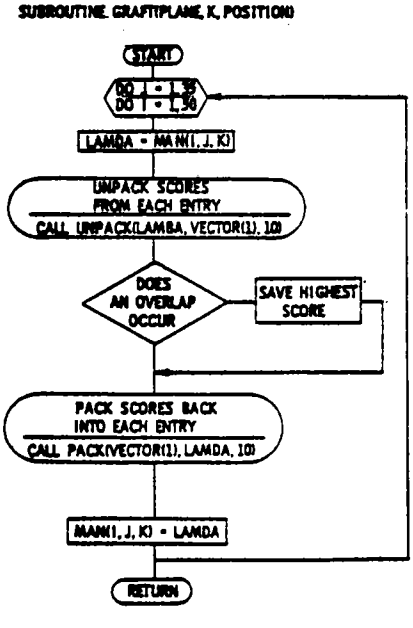
```
      TEMP=NUMCARDS+1
      K=ROW(8)*TEMP
      DO 1000 CARD=1,K
         READ(3,101)(ROW(I),I=1,25)
      KARD=KARD+1
      IF(KARD .EQ. JUMP) ROW(INDEX2)=ITEM
101      FORMAT(I3,I2,23A3)
         L=23
         IF(TEMP.EQ.ROW(2))L=RES
         L=L+2
         DO 2000 NUM=3,L
            IF(ROW(NUM).EQ.3H  8)ROW(NUM)=3H -1
            DECODE(10,107,ROW(NUM))TEMPA
107         FORMAT(I3)
            ROW(NUM)=TEMPA+1
2000     CONTINUE
         WRITE(4,106)(ROW(I),I=1,L)
106      FORMAT(I3,I2,23I3)
1000  CONTINUE
      GOTO 10
20    END FILE 4
      KOUNT=KOUNT+1
      GOTO 10
30    REWIND 3
      REWIND 4
      STOP
40    K=ROW(7)
      J=ROW(1)
      WRITE(6,102)TEMP,K,J,NUMCARDS,RES
102   FORMAT('0ITEM: ***ERROR*** CARD CHECK',/,
     1        10X,'COMPUTED NUMBER OF ENTRIES',I5,/,
     2        10X,'INPUT NUMBER OF ENTRIES',I5,2X,
     3        'ON SLICE',I5,/,10X,
     4        'COMPUTED NUMBER OF CARDS',I5,2X,
     5        'WITH',I5,2X,'LEFTOVER')
      REWIND 3
      REWIND 4
      STOP
C
C       LOOP 3000 INSERTS A NEW ROW
C
35    DO 3000 I=1,8
         ROW(I)=NEWROW(I)
3000  CONTINUE
50    READ(5,109) INDEX1,INDEX2,ITEM,JUMP,INDEX3
109   FORMAT(5I5)
      WRITE(6,110) INDEX1,INDEX2,ITEM,JUMP
110   FORMAT('0CORRECTION ENTERED FOR SLICE',I5,2X,
     1        'THE',I5,2X,'TH NUMBER SUBSTITUTED BY',I5,
     2      2X,'ON THE',I5,1X,'TH CARD OF THE SLICE')
      IF(INDEX3 .LT. 9) GOTO 25
      IF(INDEX3 .GT. 10) GOTO 45
      READ(5,108)(NEWROW(I),I=1,8)
25    CORRECTION=.FALSE.
      GOTO 60
45    FLAG=.FALSE.
      GO TO 25
70    ROW(INDEX2)=ITEM
      IF(FLAG)GO TO 50
      GOTO 80
95    RES=23
      NUMCARDS=NUMCARDS-1
      GO TO 90
      END
```

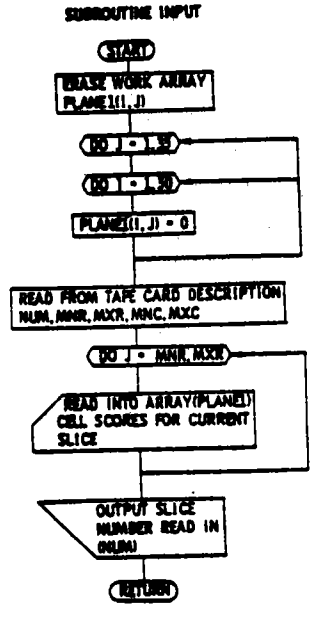
COMPUTER MAN DISC LOAD PROGRAM

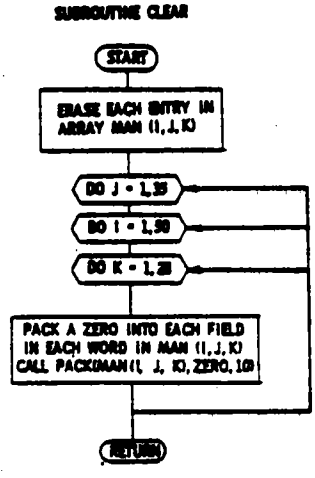


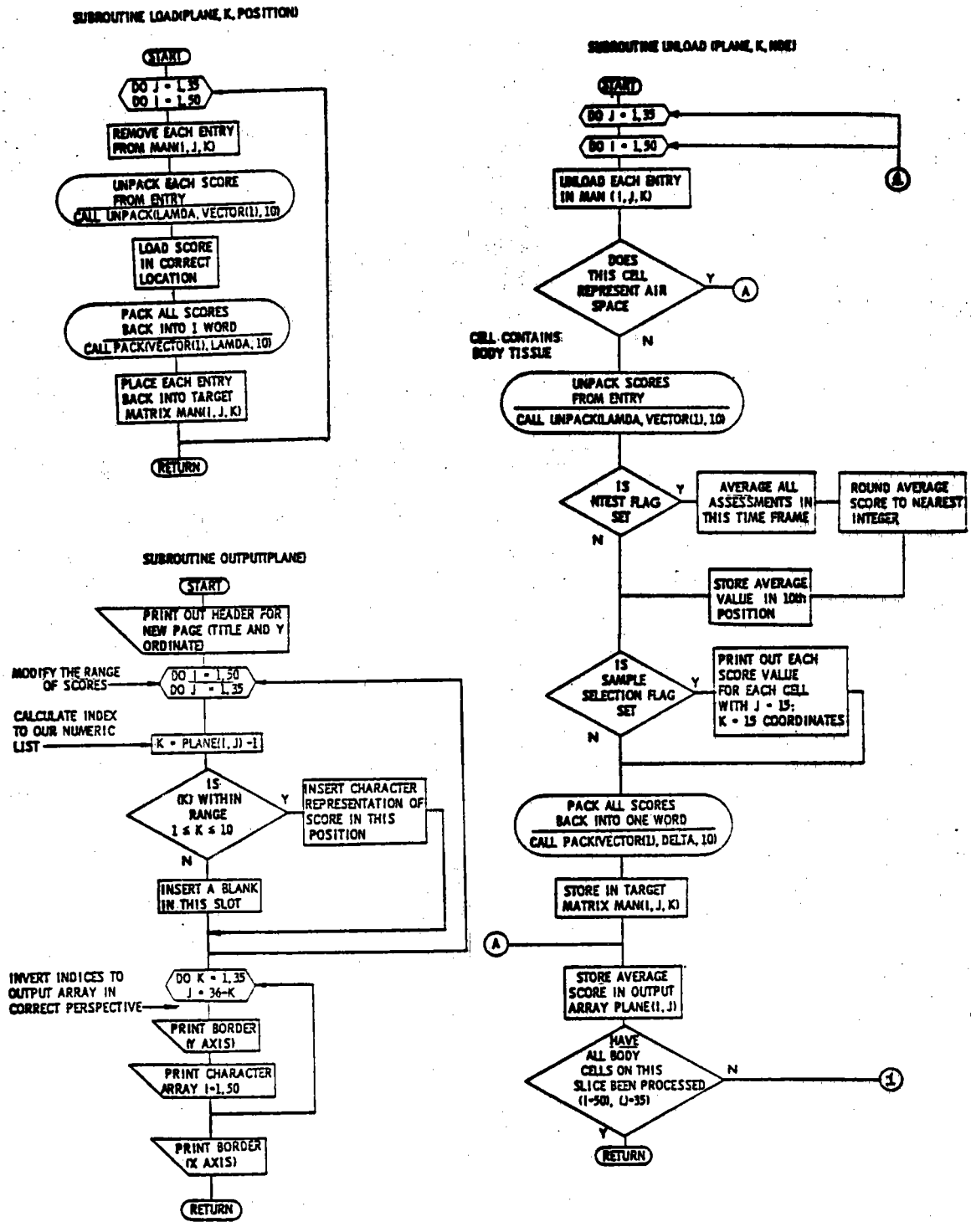

SUBROUTINE REFLECT1
START
DO J = 1,35
DO I = 1,50
ERASE WORK ARRAY PLANE1(I, J)
PLANE1(I, J) = 0
DO J = 7,35
REFLECT LEFT ARM SCORES IN PLANE 1(I, J) TO PRODUCE AN IDENTICAL SET OF RIGHT ARM SCORES
DO I = 1,25
L = I-7
PLANE2(L, J) = PLANE1(I, J)
DO I = 1,25
L = 51-I
STORE RIGHT ARM SLICES IN CORRECT POSITION
PLANE2(L, J) = PLANE1(I, J)
RETURN


SUBROUTINE REFLECT2
START
DO J = 1,35
DO I = 1,50
ERASE WORK ARRAY PLANE2 I, J)
PLANE1(I, J) = 0
DO J = 1,34
K = J + 1
SHIFT CELLS DESCRIBING LEG SLICES 3 UNITS ALONG X-AXIS, 1 UNIT ALONG Y-AXIS
DO I = 1,22
L = I + 3
PLANE2(L, K) = PLANE1(I, J)
DO J = 1,35
REFLECT LEFT LEG SCORES TO PRODUCE RIGHT LEG SCORES
K = 51 - I
PLANE1(I, J) = PLANE2(K, J)
RETURN


SUBROUTINE GRAFT(PLANE, K, POSITION)
START
DO J = 1,35
DO I = 1,50
LAMDA = MAN(I, J, K)
UNPACK SCORES FROM EACH ENTRY
CALL UNPACK(LAMDA, VECTOR(1), 10)
DOES AN OVERLAP OCCUR
SAVE HIGHEST SCORE
PACK SCORES BACK INTO EACH ENTRY
CALL PACK(VECTOR(1), LAMDA, 10)
MAN(I, J, K) = LAMDA
RETURN


SUBROUTINE INPUT
START
ERASE WORK ARRAY PLANE1(I, J)
DO J = 1,35
DO I = 1,50
PLANE1(I, J) = 0
READ FROM TAPE CARD DESCRIPTION NUM, MNR, MXR, MNC, MXC
DO J = MNR, MXR
READ INTO ARRAY(PLANE1) CELL SCORES FOR CURRENT SLICE
OUTPUT SLICE NUMBER READ IN (NUM)
RETURN


SUBROUTINE CLEAR
START
ERASE EACH ENTRY IN ARRAY MAN (I, J, K)
DO J = 1,35
DO I = 1,50
DO K = 1,28
PACK A ZERO INTO EACH FIELD IN EACH WORD IN MAN (I, J, K)
CALL PACKMAN(I, J, K), ZERO, 10)
RETURN


SUBROUTINE LOAD(PLANE, K, POSITION)
START
DO J = 1,35
DO I = 1,50
REMOVE EACH ENTRY FROM MAN(I, J, K)
UNPACK EACH SCORE FROM ENTRY
CALL UNPACK(LAMDA, VECTOR(1), 10)
LOAD SCORE IN CORRECT LOCATION
PACK ALL SCORES BACK INTO 1 WORD
CALL PACK(VECTOR(1), LAMDA, 10)
PLACE EACH ENTRY BACK INTO TARGET MATRIX MAN(I, J, K)
RETURN
SUBROUTINE UNLOAD (PLANE, K, NBE)
START
DO J = 1,35
DO I = 1,50
UNLOAD EACH ENTRY IN MAN (I, J, K)
DOES THIS CELL REPRESENT AIR SPACE
Y
A
N
CELL CONTAINS BODY TISSUE
UNPACK SCORES FROM ENTRY
CALL UNPACK(LAMDA, VECTOR(1), 10)
IS NTEST FLAG SET
Y
AVERAGE ALL ASSESSMENTS IN THIS TIME FRAME
ROUND AVERAGE SCORE TO NEAREST INTEGER
N
STORE AVERAGE VALUE IN 10th POSITION
IS SAMPLE SELECTION FLAG SET
Y
PRINT OUT EACH SCORE VALUE FOR EACH CELL WITH J = 15; K = 15 COORDINATES
N
PACK ALL SCORES BACK INTO ONE WORD
CALL PACK(VECTOR(1), DELTA, 10)
STORE IN TARGET MATRIX MAN(I, J, K)
A
STORE AVERAGE SCORE IN OUTPUT ARRAY PLANE(I, J)
HAVE ALL BODY CELLS ON THIS SLICE BEEN PROCESSED (I=50, J=35)
N
1
Y
RETURN
SUBROUTINE OUTPUT(PLANE)
START
PRINT OUT HEADER FOR NEW PAGE (TITLE AND Y ORDINATE)
MODIFY THE RANGE OF SCORES
DO I = 1,50
DO J = 1,35
CALCULATE INDEX TO OUR NUMERIC LIST
K = PLANE(I, J) +1
IS (K) WITHIN RANGE 1 ≤ K ≤ 10
Y
INSERT CHARACTER REPRESENTATION OF SCORE IN THIS POSITION
N
INSERT A BLANK IN THIS SLOT
INVERT INDICES TO OUTPUT ARRAY IN CORRECT PERSPECTIVE
DO K = 1,35
J = 36-K
PRINT BORDER (Y AXIS)
PRINT CHARACTER ARRAY I=1,50
PRINT BORDER (X AXIS)
RETURN

```
 OCT.30,76  BRLESC2 FORTRAN.
*VL779A   STANLEY   X4723 328 COMPUTER MAN DISC LOAD

*       LIST(START)
$       COMM DISC 55023 TO UNIT 3
$       COMM REEL 26C14 TO SWITCH 4 , INPUT
$       MAXT(35)MINS
$       MAXO(20000) LINES
        DIMENSION NAME(4)
        COMMON/MAN/MAN(50,35,28)
        COMMON/ PLANE1/PLANE1(50,35)
        COMMON/PLANE2/PLANE2(50,35)
        INTEGER PLANE1,PLANE2
        DATA IDD,IDF/10H55023 BODY,10HINFIN COMP/
        CALL DISCDL(3,NAME,IDD)
        CALL DISCDF(IDF,49000,400,IDUMMY,0)
        CALL CLEAR
        DO 6000 K=1,3
           CALL DISCWT(IDF,K,1,MAN)
6000    CONTINUE
        DO 1000 M=1,5
        CALL DISCRD(IDF,3,1,MAN)
        DO 2000 L=1,26
           K=27-L
           CALL INPUT
           CALL LOAD(PLANE1,K,M)
1120       FORMAT(11X,I5,10X,I5)
2000    CONTINUE
        CALL DISCWT(IDF,3,1,MAN)
        CALL DISCRD(IDF,2,1,MAN)
        DO 3000 L=27,44
           K=55-L
           CALL INPUT
           CALL LOAD(PLANE1,K,M)
3000    CONTINUE
        CALL DISCWT(IDF,2,1,MAN)
        CALL DISCRD(IDF,3,1,MAN)
        DO 7000 L=50,52
           K=53-L
           CALL INPUT
           CALL REFLECT1
           CALL GRAFT(PLANE2,K,M)
7000    CONTINUE
        CALL DISCWT(IDF,3,1,MAN)
        CALL DISCRD(IDF,2,1,MAN)
        DO 8000 L=53,75
           K=81-L
           CALL INPUT
           CALL REFLECT1
           CALL GRAFT(PLANE2,K,M)
8000    CONTINUE
        DO 9000 L=76,85
        K=86-L
           CALL INPUT
           CALL REFLECT2
           CALL GRAFT(PLANE2,K,M)
9000    CONTINUE
        CALL DISCWT(IDF,2,1,MAN)
        CALL DISCRD(IDF,1,1,MAN)
        DO 1010 L=86,113
```

```
      K=114-L
         CALL INPUT
         CALL REFLECT2
         CALL GRAFT(PLANE2,K,M)
1010  CONTINUE
      CALL DISCWT(IDF,1,1,MAN)
1000  CONTINUE
      DO 4000 M=1,5
         DO 4000 L=1,3
            CALL DISCRD(IDF,L,1,MAN)
            DO 4000 K=1,28
               CALL UNLOAD(PLANE1,K,M)
               CALL OUTPUT(PLANE1)
               WRITE(6,100)M,L,K
100            FORMAT('0DOCTOR INTERPRETATION',I5,2X,
     1            'DISC RECORD',I5,2X,'SLICE',I5,2X)
4000  CONTINUE
         REWIND 4
      STOP
      END
```

```
*     LIST(START)
C
C
C
      SUBROUTINE CLEAR
      DIMENSION ZERO(10)
      COMMON/MAN/MAN(50,35,28)
      INTEGER ZERO
      DATA ZERO/10*0/
      DO 1000 J=1,35
         DO 1000 I=1,50
            DO 1000 K=1,28
            CALL PACK(MAN(I,J,K),ZERO,10)
1000  CONTINUE
      RETURN
      END
```

```
*      LIST(START)
C
C
C
       SUBROUTINE REFLECT1
       COMMON/PLANE2/PLANE2(50,35)
       COMMON/ PLANE1/PLANE1(50,35)
       INTEGER PLANE1,PLANE2
       DO 1000 J=1,35
          DO 1000 I=1,50
             PLANE2(I,J)=0
1000   CONTINUE
       DO 2000 J=7,35
             DO 3000 I=8,25
                L=I-7
                PLANE2(L,J)=PLANE1(I,J)
3000         CONTINUE
             DO 4000 I=1,25
                L=51-I
                PLANE2(L,J)=PLANE2(I,J)
4000         CONTINUE
2000   CONTINUE
       RETURN
       END
```

```
*      LIST(START)
C
C
C
       SUBROUTINE REFLECT2
       COMMON/PLANE2/PLANE2(50,35)
       COMMON/ PLANE1/PLANE1(50,35)
       INTEGER PLANE1,PLANE2
       DO 1000 J=1,35
          DO 1000 I=1,50
             PLANE2(I,J)=0
1000   CONTINUE
       DO 2000 J=1,34
          K=J+1
          DO 2000 I=1,22
             L=I+3
             PLANE2(L,K)=PLANE1(I,J)
2000   CONTINUE
       DO 3000 J=1,35
          DO 3000 I=26,50
             K=51-I
             PLANE2(I,J)=PLANE2(K,J)
3000   CONTINUE
       RETURN
       END
```

```
*       LIST(START)
C
C
C
        SUBROUTINE GRAFT(PLANE,K,POSITION)
        INTEGER PLANE,POSITION,VECTOR
        DIMENSION PLANE(50,35)
        DIMENSION VECTOR(10)
        COMMON/MAN/MAN(50,35,28)
        DO 1000 J=1,35
           DO 1000 I=1,50
              LAMDA=MAN(I,J,K)
                CALL UNPACK(LAMDA,VECTOR(1),10)
              IF(VECTOR(POSITION).LT.PLANE(I,J))
     *              VECTOR(POSITION)=PLANE(I,J)
              CALL PACK(VECTOR(1),LAMDA,10)
              MAN(I,J,K)=LAMDA
1000    CONTINUE
        RETURN
        END
```

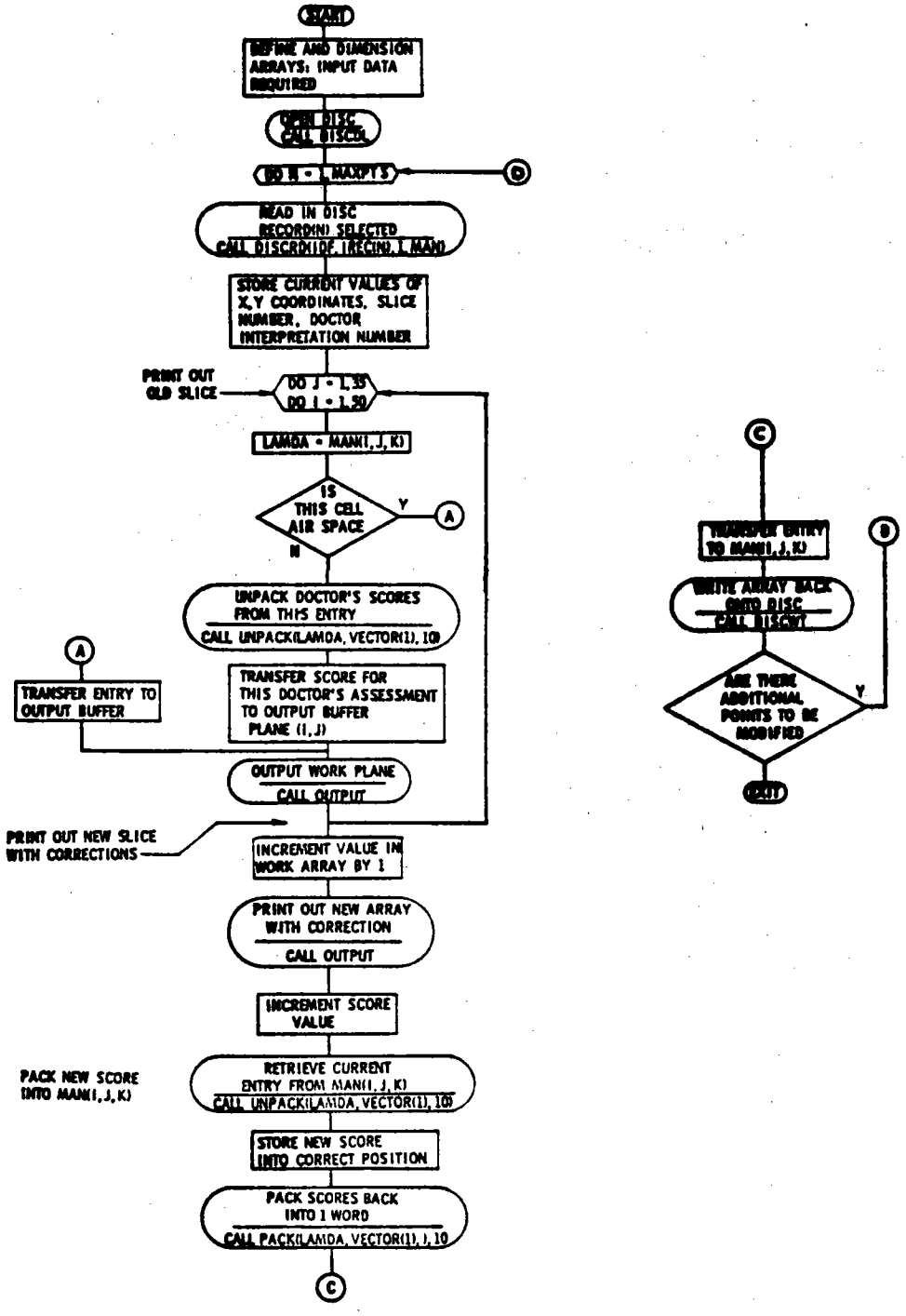
```
*       LIST(START)
C
C
C
        SUBROUTINE LOAD(PLANE,K,POSITION)
        DIMENSION PLANE(50,35)
        INTEGER PLANE,POSITION,VECTOR
        DIMENSION VECTOR(10)
        COMMON/MAN/MAN(50,35,28)
        DO 1000 J=1,35
           DO 1000 I=1,50
              LAMDA=MAN(I,J,K)
              CALL UNPACK(LAMDA,VECTOR(1),10)
              VECTOR(POSITION)=PLANE(I,J)
              CALL PACK(VECTOR(1),LAMDA,10)
              MAN(I,J,K)=LAMDA
1000    CONTINUE
        RETURN
        END
```

```
*       LIST(START)
C
C
C
      SUBROUTINE OUTPUT(PLANE)
      DIMENSION PLANE(50,35)
      DIMENSION PLANE2(50,35)
      DIMENSION NUM(10)
      DIMENSION BORDER(100)
      INTEGER BORDER
      INTEGER PLANE
      INTEGER PLANE2
      INTEGER CHAR
      INTEGER BLANK
      DATA BORDER/100*1H./
      DATA CHAR/' .'/
      DATA NUM/' 1',' 2',' 3',' 4',' 5',' 6',
     1' 7',' 8',' 9',' 0'/
      DATA BLANK/'  '
      WRITE(6,101)
101   FORMAT('1NEW PLANE',//,' ',//,' +Y')
      DO 2000 I=1,50
      DO 2000 J=1,35
      K=PLANE(I,J)+1
      IF(K.GE.1.AND.K.LE.10) GO TO 105
      PLANE2(I,J)=BLANK
      GO TO 2000
105   PLANE2(I,J)=NUM(K)
2000  CONTINUE
      DO 1000 K=1,35
         J=36-K
      WRITE(6,100) CHAR,(PLANE2(I,J),I=1,50)
100   FORMAT(2X,1A2,50A2)
1000  CONTINUE
      WRITE(6,103)(BORDER(I),I=1,50)
103   FORMAT(4X,50A2)
      WRITE(6,102)
102   FORMAT(' (0,0)',95X,' +X =>')
      RETURN
      END

C
      SUBROUTINE INPUT
C      THIS SUBROUTINE READS THE CARD DESCRIPTION OF THE ARRAY MAN AND PLACES
C     INTO ACTIVE MEMORY IN PLANE FOR MANIPULATION AND ASSIGNMENT INTO MAN
      COMMON/ PLANE1/PLANE1(50,35)
      INTEGER PLANE1
      DO 1000 J=1,35
         DO 1000 I=1,50
            PLANE1(I,J)=0
1000  CONTINUE
         READ(4,100)NUM,MNR,MXR,MNC,MXC
100      FORMAT(I3,2X,4I4)
         DO 2000 J=MNR,MXR
            READ(4,101)(PLANE1(I,J),I=MNC,MXC)
101         FORMAT(5X,23I3)
2000  CONTINUE
      WRITE(6,103)NUM
103   FORMAT('0PLANE',I3,2X,' READ IN')
      RETURN
      END
```

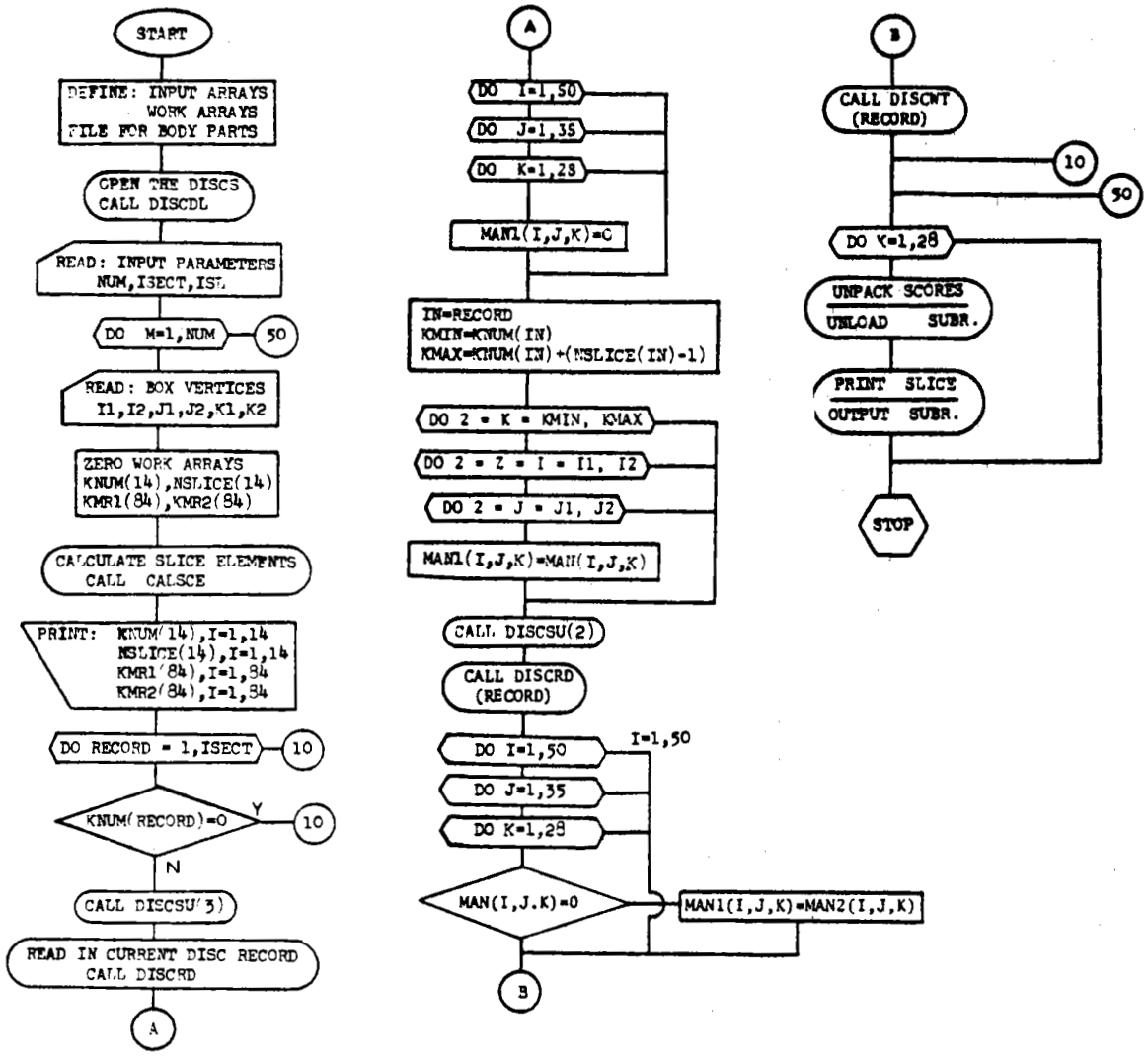
```
*       LIST(START)
C
C
C
        SUBROUTINE UNLOAD(PLANE,K,NDE)
        INTEGER PLANE,POSITION,VECTOR
        INTEGER DELTA
        DIMENSION PLANE(50,35)
        DIMENSION VECTOR(10)
        COMMON/MAN/MAN(50,35,28)
        COMMON KCHECK
        DO 1000 J=1,35
           DO 1000 I=1,50
              LAMDA=MAN(I,J,K)
C       IS THIS CELL AIR SPACE
C       OR IS IT PART OF THE MAN
        IF(LAMDA.EQ.0) GO TO 1060
              CALL UNPACK(LAMDA,VECTOR(1),10)
        KNDE=NDE
        IF(KCHECK.EQ.1) KNDE=NDE-1
C       AVERAGE CELL VALUES
C       ROUND   CELL  VALUES TO  NEAREST  INTEGER
        ITOTAL=0
        DO 1010 L=1,KNDE
        ITOTAL=ITOTAL+VECTOR(L)
1010    CONTINUE
        ATOTAL=FLOAT(ITOTAL)
        SNDE=FLOAT(KNDE)
        AVG1=ATOTAL/SNDE
        LAMDA=IFIX(AVG1)
        BETA=FLOAT(LAMDA)
        RES=AVG1-BETA
        IF(RES.GT.0.49) LAMDA=LAMDA+1
C       PROGRAM AVERAGES SCORES AND PACKS AN AVERAGE SCORE INTO
C       EACH WORD IN 10TH POSITION
        VECTOR(10)=LAMDA
        IF(K-15)1020,1030,1020
1030    IF(J-15)1020,1040,1020
1040    WRITE(6,1050)(VECTOR(N),N=1,10),ITOTAL,ATOTAL,SNDE,AVG1,LAMDA,
       1BETA,RES
1050    FORMAT(5X,10I3,3X,I4,3X,3F5.2,3X,I3,3X,F5.2,3X,F5.2)
C       PACK ALL SCORES BACK INTO ONE WORD
1020    CALL PACK(VECTOR(1),DELTA,10)
        MAN(I,J,K)=DELTA
1060    PLANE(I,J)=LAMDA
1000    CONTINUE
        RETURN
        END
*       LIST
*       DATA
```

COMPUTER MAN CHANGE VALUE PROGRAM

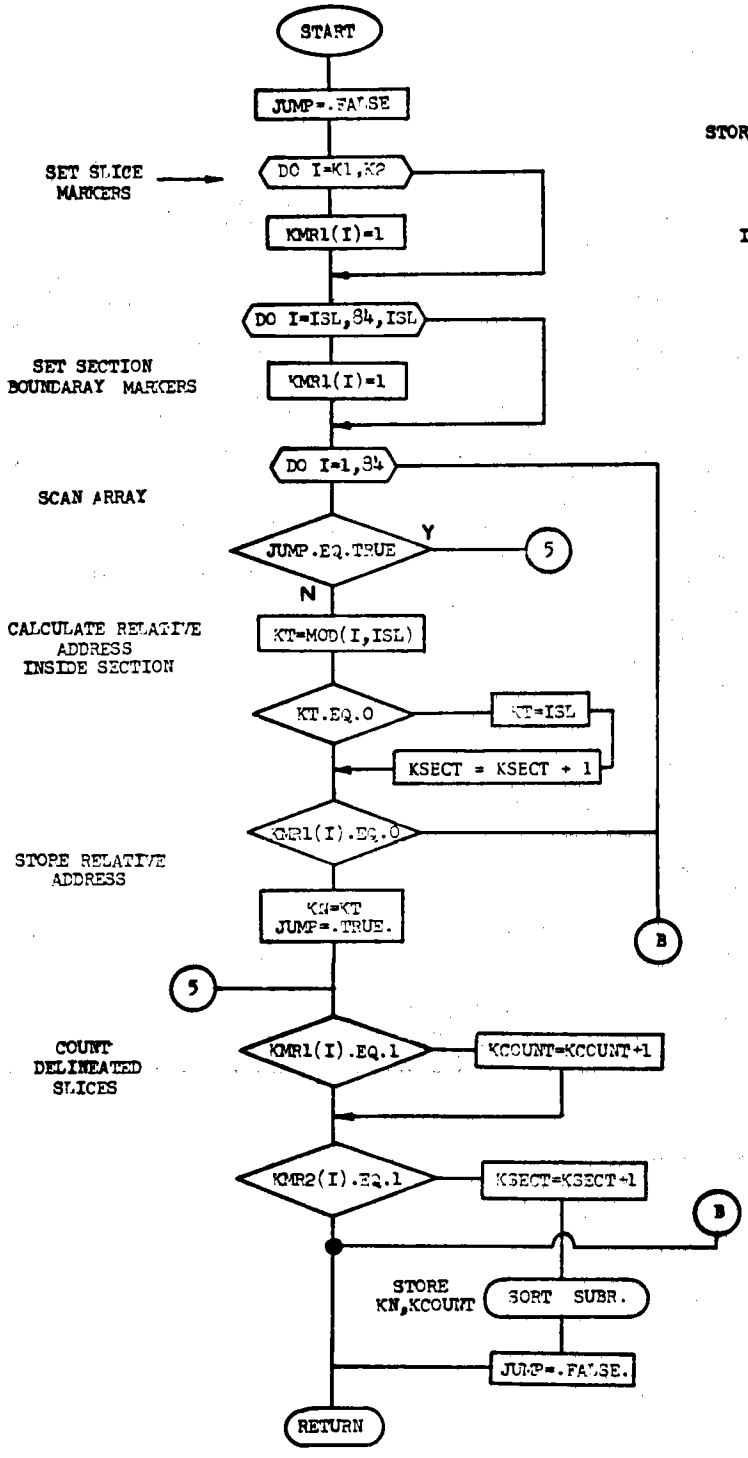


```
      LIST(START)
$     COMM DISC 50C21 TO UNIT 3
$     MAXT(25)MINS
$     MAXO(15000)LINES
      DIMENSION VECTOR(10)
      DIMENSION NAME(4),ISLICE(10),IX(10),IY(10),NEWVAL(10),IREC(10),
     1DOCINT(10)
      COMMON/MAN/MAN(50,35,28)
      COMMON/ PLANE1/PLANE1(50,35)
      COMMON/PLANE2/PLANE2(50,35)
      DIMENSION PLANE(50,35)
      INTEGER PLANE,DOCINT
      INTEGER PLANE1,PLANE2
      INTEGER VECTOR
      DATA MAXPTS/3/
      DATA DOCINT/3,2,2,7*0/
      DATA IREC/3,3,3,7*0/
      DATA ISLICE/2,1,1,7*0/
      DATA NEWVAL/9,1,1,7*0/
      DATA IX/27,13,40,7*0/
      DATA IY/17,15,15,7*0/
      DATA IDD,IDF/10H5CD21 BODY,10HBODY      /
C        NUMBER OF POINTS TO BE CHANGED
C        INPUT X,Y,Z COORDINATES ,SLICE*,NEW VALUE
C     DOCTOR INTERPRETATION NUMBER
         CALL DISCDL(3,NAME,IDD)
      DO 100 N=1,MAXPTS
C     UTILIZING DATA INPUT READ IN CORRECT DISC RECORD
      CALL DISCRD (IDF,IREC(N),1,MAN)
C        STORE SLICE NUMBER X,Y,COORDINATES
      II=IX(N)
      IJ=IY(N)
      K=ISLICE(N)
      L=DOCINT(N)
C        RETRIEVE CORRECT ENTRY FROM MAN(I,J,K)
      DO  1000 J=1,35
      DO  1000 I=1,50
      LAMDA=MAN(I,J,K)
      IF(LAMDA.EQ.0) GO TO 1020
      CALL UNPACK(LAMDA,VECTOR(1),10)
      PLANE(I,J)=VECTOR(L)
      GO TO 1000
1020  PLANE(I,J)=LAMDA
1000  CONTINUE
C     PRINT OUT OLD ARRAY
      CALL OUTPUT(PLANE)
C     PRINT OUT NEW ARRAY WITH CORRECTIONS
      PLANE(II,IJ)=NEWVAL(N)+1
      CALL OUTPUT(PLANE)
C     INCREMENT VALUE, STORE ON DISC
      NEWVAL(N)=NEWVAL(N)+1
C     PACK NEW VALUE INTO MAN(I,J,K)
      LAMDA=MAN(II,IJ,K)
      CALL UNPACK(LAMDA,VECTOR(1),10)
      VECTOR(L)=NEWVAL(N)
      CALL PACK(VECTOR(1),LAMDA,10)
      MAN(II,IJ,K)=LAMDA
      CALL DISCWT(IDF,IREC(N),1,MAN)
  100 CONTINUE
```

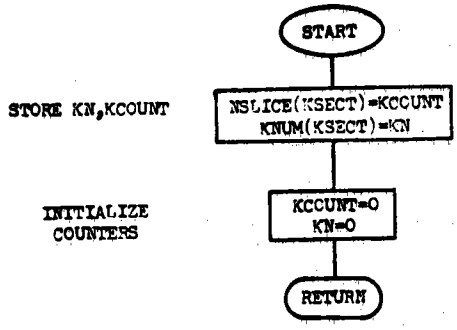
COMPUTER MAN DELINEATION PROGRAM



SUBROUTINE CALSCE



SUBROUTINE INSERT



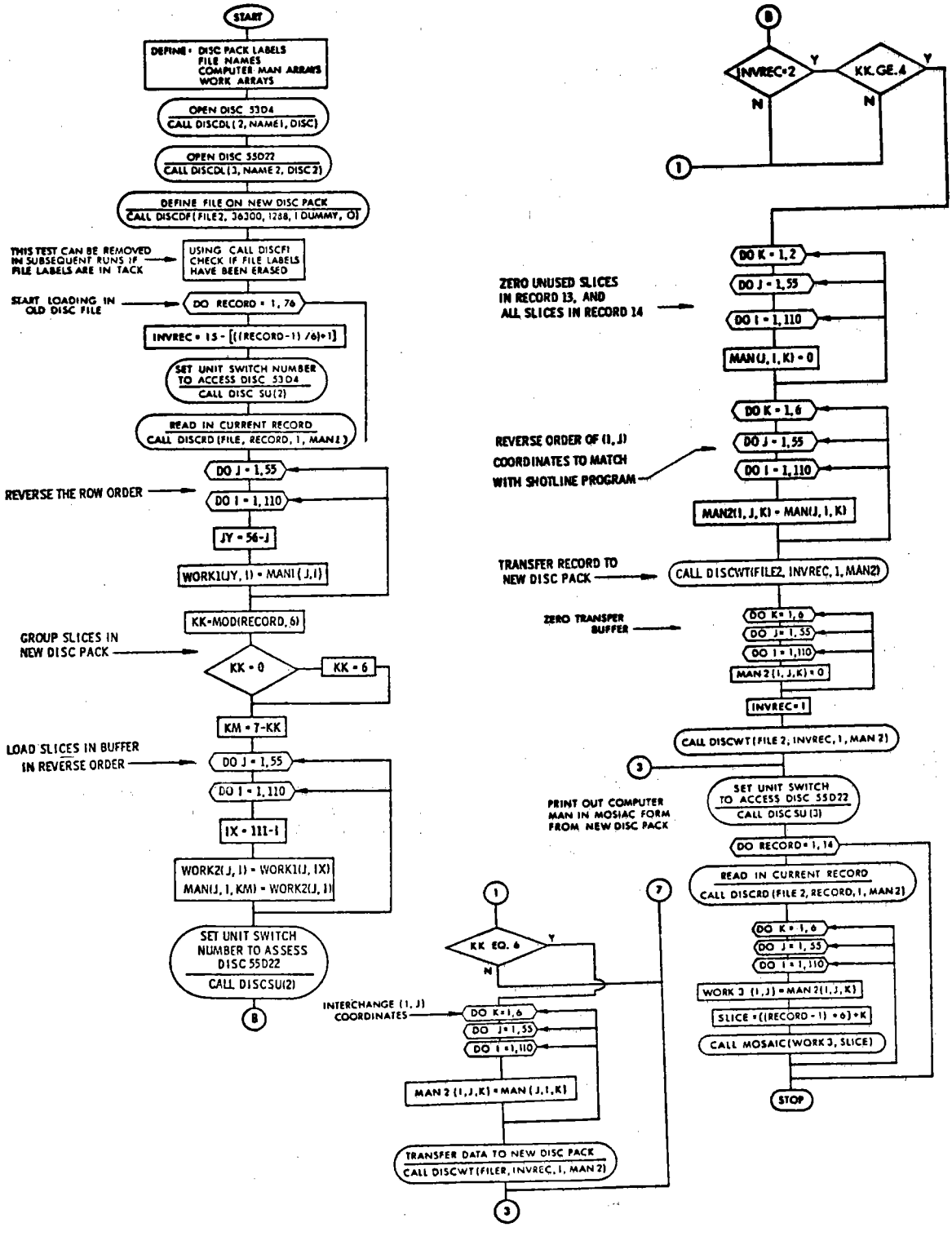
```
      COMMON/MAN/MAN(11C,55,6),NSLICE(14),KNUM(14),KMR1(84),KMR2(84),I1,
     1I2,J1,J2,K1,K2,ISECT,ISL,NSECT,KCOUNT,KSECT,KT,KMIN,KMAX,KN
      COMMON/MAN1/MAN1(11C,55,6)
      COMMON/MAN2/MAN2(11C,55,6)
      COMMON/WORK3/WORK3(11C,55)
      COMMON/NAME2/NAME2(4)
      COMMON/NAME3/NAME3(4)
      INTEGER DISC2,FILE2,WORK3
      INTEGER DISC3,FILE3
      INTEGER RECORD
      INTEGER SLICE
      DATA DISC2/1CH55D22 CMAN/
      DATA DISC3/1CH55D24 KMAN/
      DATA  FILE2/1CHINCAP BODY/
      DATA FILE3/1CHBODY PARTS/
      CALL DISCDL(3,NAME2,DISC2)
      CALL DISCDL(2,NAME3,DISC3)
C     DEFINE FILE FOR NEW DISC PACK
      CALL DISCDF(FILE3,36300,1200,IDUMMY,0)
C     NUM IS THE NUMBER OF SEPERATE INTERIOR BOXES REQUIRED TO SEPERATE
C     A DESIGNATED SECTION FROM THE REMAINDER OF THE COMPUTER MAN
      READ(5,120)NUM,ISECT,ISL
  120 FORMAT(3I3)
      CALL DISCSU(2)
*     CO=1
      DO 5C M=1,NUM
      READ(5,1CC) I1,I2,J1,J2,K1,K2
  100 FORMAT(6I3)
      DO 21 I=1,ISECT
      KNUM(I)=0
      NSLICE(I)=0
   21 CONTINUE
      DO 22 I=1,84
      KMR1(I)=0
      KMR2(I)=0
   22 CONTINUE
      CALL CALSCE
      WRITE(6,110)
110   FORMAT(10X,'KNUM',10X,'NSLICE')
      DO 7 I=1,14
      WRITE(6,1C2)(KNUM(I),NSLICE(I))
7     CONTINUE
102   FORMAT(10X,I4,10X,I4)
      WRITE(6,1C3)
103   FORMAT(10X,'KMR1',10X,'KMR2')
      DO 8 I=1,84
      II=85-I
      WRITE(6,104) (KMR1(II)),(KMR2(II))
8     CONTINUE
104   FORMAT(10X,I4,10X,I4)
      DO 1C RECORD=1,ISECT
      IF(KNUM(RECORD).EQ.0) GO TO 1C
      CALL DISCSU(3)
      CALL DISCRD(FILE2,RECORD,1,MAN)
C     ZERO ARRAY MAN1
      DO 11 I=1,11C
      DO 11 J=1,55
      DO 11 K=1,6
      MAN1(I,J,K)=C
11    CONTINUE
      IN=RECORD
      KMIN=KNUM(IN)
      KMAX=KNUM(IN)+(NSLICE(IN)-1)
      DO 2 K=KMIN,KMAX
      DO 2 I=I1,I2
      DO 2 J=J1,J2
      MAN1(I,J,K)=MAN(I,J,K)
2     CONTINUE
```

```
      CALL  DISCSU(2)
C     ADD THE FOLLOWING INSTRUCTIONS WHEN MORE THAN 1 INTERIOR BOXIS
C     REQUIRED TO SECTION OFF THE DESIRED PORTION OF BODY
C     OTHERWISE OMIT
      CALL DISCRD(FILE3,RECORD,1,MAN2)
      DO 20 I=1,110
      DO 20 J=1,55
      DO 20 K=1,6
      IF(MAN1(I,J,K).EQ.0) MAN1(I,J,K)=MAN2(I,J,K)
   20 CONTINUE
      CALL DISCWT(FILE3,RECORD,1,MAN1)
10    CONTINUE
   50 CONTINUE
C
C
C     PRINT OUT COMPUTER MAN IN MOSAIC REPRESENTATION
      DO 60 RECORD=1,ISECT
      CALL DISCRD(FILE3,RECORD,1,MAN2)
      DO 18 K=1,6

      DO 16 J=1,55
      DO 16 I=1,110
      WORK3(I,J)=MAN2(I,J,K)
16    CONTINUE
C     CALCULATE CROSS SECTION NUMBER (1-84)
      SLICE=(RECORD-1)*6+K
      CALL MOSAIC(WORK3,SLICE)
18    CONTINUE
   60 CONTINUE
      END
```

```
      SUBROUTINE CALSCE
      COMMON/MAN/MAN(110,55,6),NSLICE(14),KNUM(14),KMR1(84),KMR2(84),I1,
     1I2,J1,J2,K1,K2,ISECT,ISL,NSECT,KCOUNT,KSECT,KT,KMIN,KMAX,KN
      LOGICAL JUMP
C     JUMP = TRUE INDICATES THAT THE 1ST DELINEATED SLICE IN A SECTION
C     HAS BEEN ENCOUNTERED
C     SET SECTION MARKERS
C     SET SLICE MARKERS
      JUMP=.FALSE.
      KSECT=0
      DO 2 I=K1,K2
      KMR1(I)=1
    2 CONTINUE
      DO 1 I=ISL,84,ISL
      KMR2(I)=1
    1 CONTINUE
      DO 3 I=1,84
      IF(JUMP) GO TO 5
      KT=MOD(I,ISL)
C     WHERE KT VARIES FROM 1-ISL
      IF(KT.EQ.0) GO TO 6
    8 IF(KMR1(I).EQ.0) GO TO 3
      KN=KT
      JUMP=.TRUE.
    5 IF(KMR1(I).EQ.1) KCOUNT=KCOUNT+1
      IF(KMR2(I).EQ.1) GO TO 4
      GO TO 3
    4 KSECT=KSECT+1
      CALL SORT
      JUMP=.FALSE.
    3 CONTINUE
      GO TO 7
    6 KT=ISL
      IF(KMR1(I).EQ.0) KSECT=KSECT+1
      GO TO 8
    7 RETURN
      END


      SUBROUTINE SORT
C     KSECT REPRESENTS THE NUMBER OF SECTIONS
C     THAT CONTAIN DELINEATED SLICES .
      COMMON/MAN/MAN(110,55,6),NSLICE(14),KNUM(14),KMR1(84),KMR2(84),I1,
     1I2,J1,J2,K1,K2,ISECT,ISL,NSECT,KCOUNT,KSECT,KT,KMIN,KMAX,KN
C     SAVE ADDRESS WHERE THE 1ST 1 APPEARS RELATIVE TO THE
C     BEGINNING OF EACH SECTION
      NSLICE(KSECT)=KCOUNT
      KNUM(KSECT)=KN
      KCOUNT=0
      KN=0
      RETURN
      END
```

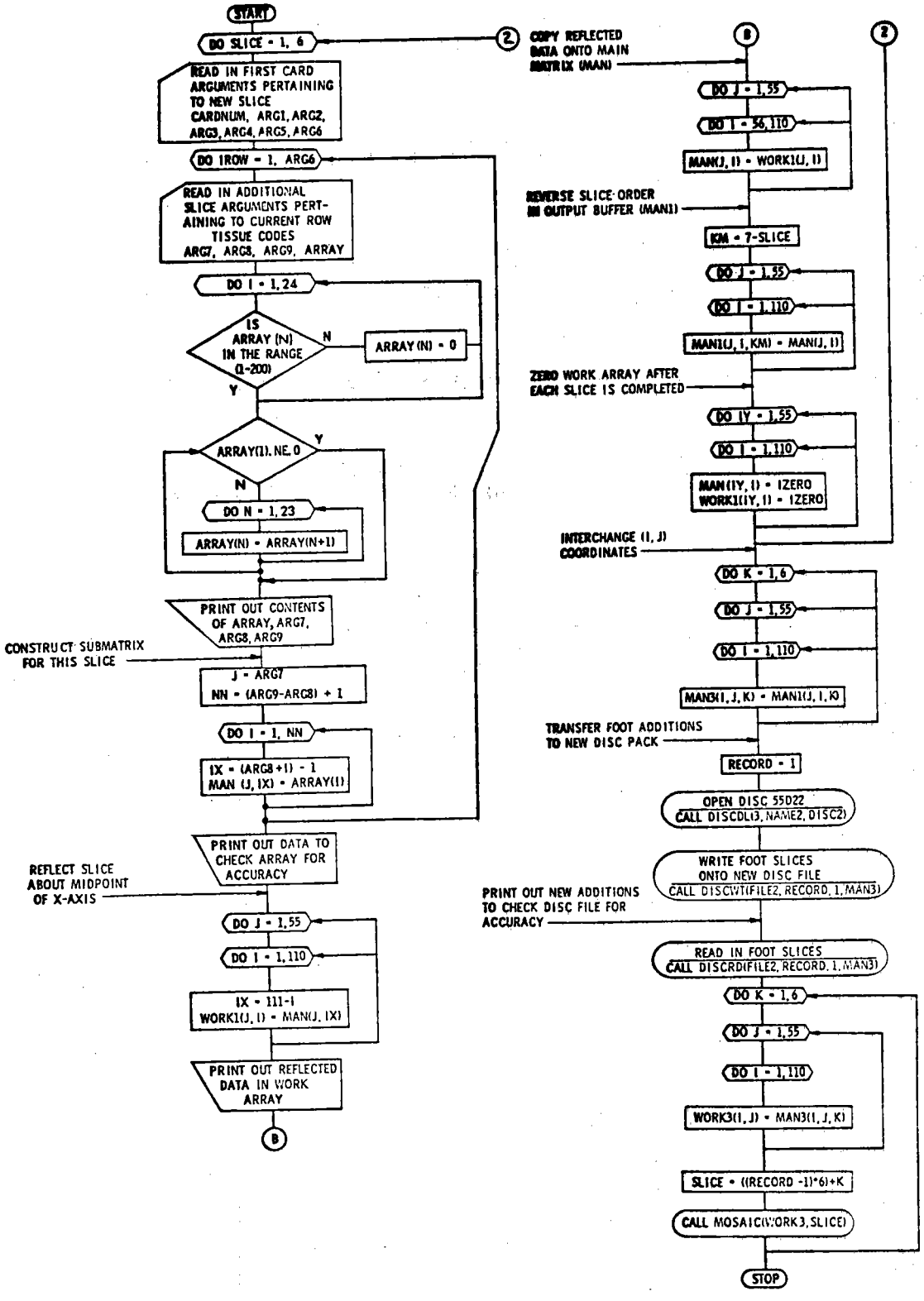

START
DEFINE: DISC PACK LABELS
FILE NAMES
COMPUTER MAN ARRAYS
WORK ARRAYS
OPEN DISC 53D4
CALL DISCDL(2, NAME1, DISC)
OPEN DISC 55D22
CALL DISCDL(3, NAME 2, DISC2)
DEFINE FILE ON NEW DISC PACK
CALL DISCDF(FILE2, 36300, 1288, 1 DUMMY, 0)
THIS TEST CAN BE REMOVED IN SUBSEQUENT RUNS IF FILE LABELS ARE IN TACK
USING CALL DISCFI CHECK IF FILE LABELS HAVE BEEN ERASED
START LOADING IN OLD DISC FILE
DO RECORD = 1, 76
INVREC = 15 - [((RECORD-1) /6)+1]
SET UNIT SWITCH NUMBER TO ACCESS DISC 53D4
CALL DISC SU(2)
READ IN CURRENT RECORD
CALL DISCRD (FILE, RECORD, 1, MAN2)
REVERSE THE ROW ORDER
DO J = 1,55
DO I = 1,110
JY = 56-J
WORK1(JY, I) = MAN1(J,I)
KK=MOD(RECORD, 6)
GROUP SLICES IN NEW DISC PACK
KK = 0
KK = 6
KM = 7-KK
LOAD SLICES IN BUFFER IN REVERSE ORDER
DO J = 1,55
DO I = 1,110
IX = 111-I
WORK2(J, I) = WORK1(J, IX)
MAN(J, I, KM) = WORK2(J, I)
SET UNIT SWITCH NUMBER TO ASSESS DISC 55D22
CALL DISCSU(2)
B
INVREC=2
Y
N
KK.GE.4
Y
N
1
ZERO UNUSED SLICES IN RECORD 13, AND ALL SLICES IN RECORD 14
DO K = 1,2
DO J = 1,55
DO I = 1,110
MAN(J, I, K) = 0
REVERSE ORDER OF (I, J) COORDINATES TO MATCH WITH SHOTLINE PROGRAM
DO K = 1,6
DO J = 1,55
DO I = 1,110
MAN2(I, J, K) = MAN(J, I, K)
TRANSFER RECORD TO NEW DISC PACK
CALL DISCWT(FILE2, INVREC, 1, MAN2)
ZERO TRANSFER BUFFER
DO K= 1,6
DO J= 1,55
DO I = 1,110
MAN 2 (I, J,K) = 0
INVREC= 1
CALL DISCWT(FILE 2; INVREC, 1, MAN 2)
3
PRINT OUT COMPUTER MAN IN MOSIAC FORM FROM NEW DISC PACK
SET UNIT SWITCH TO ACCESS DISC 55D22
CALL DISC SU (3)
DO RECORD= 1, 14
READ IN CURRENT RECORD
CALL DISCRD (FILE 2, RECORD, 1, MAN2)
DO K = 1, 6
DO J = 1, 55
DO I = 1,110
WORK 3 (I, J) = MAN 2(I, J, K)
SLICE = ((RECORD - 1) = 6) + K
CALL MOSAIC(WORK 3, SLICE)
STOP
1
KK EQ. 6
Y
N
INTERCHANGE (I, J) COORDINATES
DO K=1,6
DO J=1,55
DO I =1,110
MAN 2 (I, J, K) = MAN (J, I, K)
TRANSFER DATA TO NEW DISC PACK
CALL DISCWT (FILER, INVREC, 1, MAN 2)
7
3

```
*      LIST(START)
$      MAXO(5000)LINES
$      MAXT(18)MINS
$      COMM DISC 53D4 TO UNIT 2
$      COMM DISC 55D22 TO UNIT 3
C
C                              MAIN ROUTINE
C      THIS PROGRAM IS DESIGNED TO TRANSFORM THE POSITION OF THE
C      COMPUTER MAN TO MATCH THAT OF THE LETHALITY DESCRIPTION
C      ALSO THE RECORD LENGTH IS INCREASED FROM 1-6 SLICES PER RECORD
C      OUR NEW COMPUTER MAN IS STORED ON DISC PACK 55D22
C
       DATA DISC/10H53D4 C MAN/
       DATA DISC2/10H55D22 CMAN/
       DATA FILE/10HBODY      /
       DATA FILE2/10HINCAP BODY/
         DATA READER/5/,WRITER/6/
       COMMON /MAN/MAN(55,110,6)
       COMMON/MAN1/MAN1(55,110)
       COMMON/MAN2/MAN2(110,55,6)
       COMMON /WORK1/WORK1(55,110)
       COMMON /WORK2/WORK2(55,110)
       COMMON/WORK3/WORK3(110,55)
       COMMON /NAME1/NAME1(4)
       COMMON /NAME2/NAME2(4)

       DIMENSION IFIL(131),LOR(131),ITR(131),LTR(131),MAXNR(131),
      AIHRU(131),IHTU(131),ICI(131)
       INTEGER WORK1,WORK2,RECORD
       INTEGER FILE,FILE2
       INTEGER WORK3
       INTEGER    SLICE
       INTEGER READER,WRITER,DISC,DISC2
C      CM INCAPACTATION MODEL IS DIVIDED INTO 14 RECORDS
C      SIX SLICES PER RECORD
C      THE RECORD LENGTH FOR FILE 2 IS 36300 WORDS
C
C      OPEN THE DISC TO BOTH FILES 53D4 ,55D22
C      THE RECORD LENGTH OF FILE1 IS 6050 WORDS
C      CALCULATE CORRECT RELATIVE RECORD NUMBER FOR NEW FILE
C       FILE 1    1-6    7-12  13-18        79-84
C                 ---    ----  ----- ..... -----
C       FILE 2     1       2     3            14
       CALL DISCDL(2,NAME1,DISC)
       CALL DISCDL(3,NAME2,DISC2)
       CALL DISCDF(FILE2,36300,1288,IDUMMY,0)
       IFIL(1)=0.0
       CALL DISCFI(IFIL,LOR,ITR,LTR,MAXNR,IHRU,IHTU,ICI)
       DO 328 I=1,131
       IF(IFIL(I).NE.0) WRITE(6,110) (IFIL(I),LOR(I),ITR(I),LTR(I),
      AMAXNR(I),IHRU(I),IHTU(I),ICI(I))
328    CONTINUE
110    FORMAT(A10,2X,7(I4,2X))
       DO 11 RECORD=1,76
       INVREC=((RECORD-1)/6)+1
       INVREC=15-INVREC
C      LOAD HIGHEST RECORD FIRST
```

```
      CALL DISCSU(2)
      CALL DISCRD(FILE,RECORD,1,MAN1)
C     K IS EQUAL TO SLICE NUMBER WITHIN RECORD
C     INVERT COMPUTER MAN
      DO 20 J=1,55
      DO 20 I=1,110
      JY=56-J
      WORK1(JY,I)=MAN1(J,I)
20    CONTINUE
C     1ST ROTATE CM 180 DEGREES CLOCKWISE ABOUT X-AXIS
C     2ND ROTATE CM 180 DEGREES COUNTERCLOCKWISE ABOUT Z AXIS
C     ROTATE COMPUTER MAN 90DEGREES COUNTERCLOCKWISE ABOUT Z-AXIS
C     REVERSE THE ROW ORDER
C     REVERSE THE COLUMN ORDER
C     REVERSE THE RECORD ORDER
C     LOAD BACK INTO MAN
      KK=MOD(RECORD,6)
      IF(KK.EQ.0) KK=6
C     REVERSE SLICE ORDER BEFORE LOADING INTO NEW RECORD
      KM=7-KK
C     PERFORM A LIMIT CHECK ON  KM
      IF(KM.GE.1.AND.KM.LE.6) GO TO 71
      WRITE(6,108) KM
108   FORMAT(1 X,I4)
      GO TO  11
71    CONTINUE
      DO 30 J=1,55
      DO 30 I=1,110
      IX=111-I
      WORK2(J,I)=WORK1(J,IX)
      MAN(J,I,KM)=WORK2(J,I)
30    CONTINUE
C     TRANSFER TO NEW DISC FILE
      CALL DISCSU(3)
C     DEFINE DISC FILES FOR NEW DISC PACK
C     FILE2 IS NEW DISC LABEL
C     CALL DISCDF(IFIL,LOR,NT,IAV,K)
C     NT IS NUMBER OF TRACKS NEEDED 36300/398
C     LOR IS LENGTH OF DISC RECORD
C     IAV IS ADUMMMY VARIABLE
C     K DEFINES SOME ACTION TO BE TAKEN ON IFILE  K=0 CLEARS FILE
C     IF RECORD FILE FILE2 IS FILLED WRITE RECORD ONTO DISC
C
C
C     WRITE ON UNWRITTEN RECORD 1 AND RECORD 2 SLICES 5,6
      IF(INVREC.EQ.2.AND.KK.GE.4) GO TO 50
      GO TO 12
50    DO 80 K=1,2
      DO 80 J=1,55
      DO 80 I=1,110
      MAN(J,I,K)=0
80    CONTINUE
      DO 3 K=1,6
      DO 3 J=1,55
      DO 3 I=1,110
      MAN2(I,J,K)=MAN(J,I,K)
3     CONTINUE
      CALL DISCWT(FILE2,INVREC,1,MAN2)
      GO TO 70
```

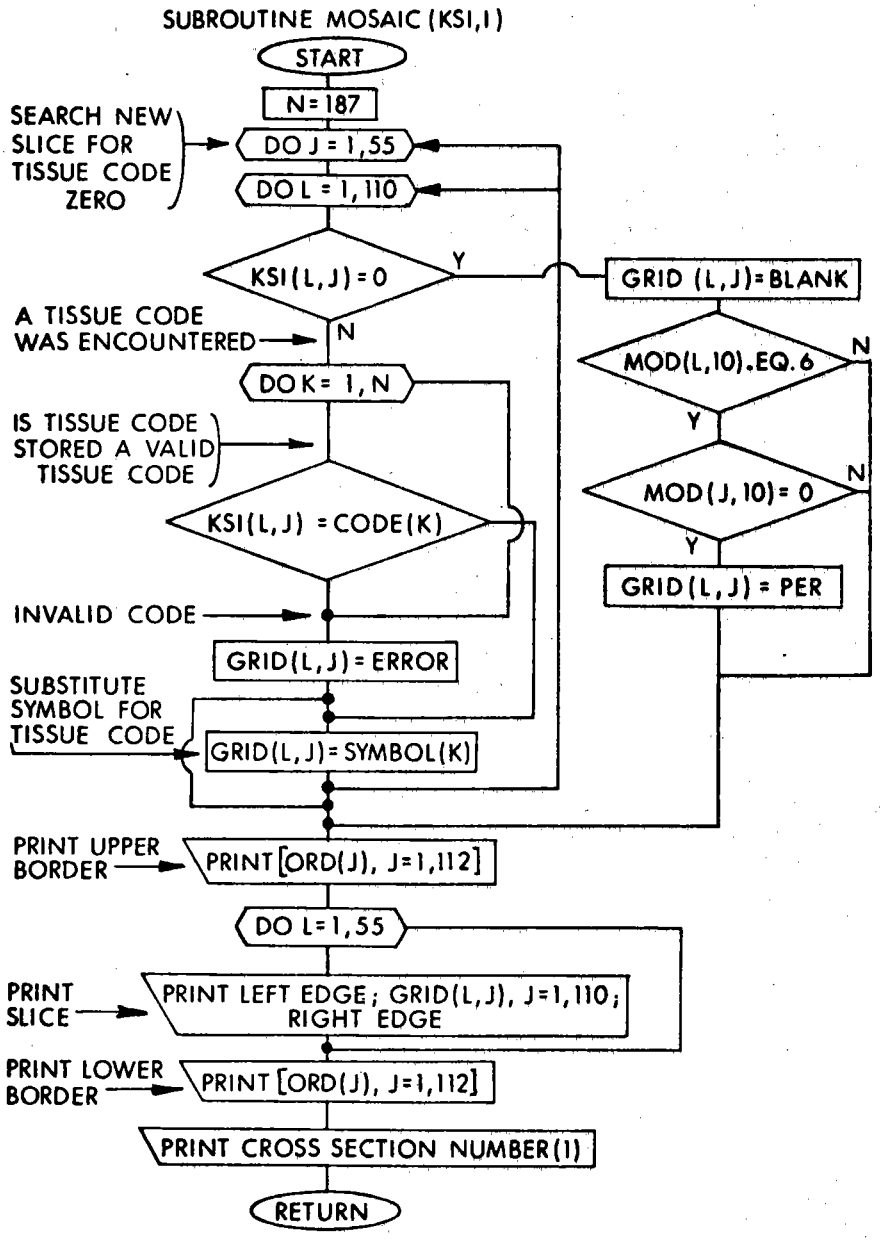
```
12      CONTINUE
        IF(KK.EQ.6)GO TO 101
11      CONTINUE
101     CONTINUE
C
C
C       INTERCHANGE X-Y COORDINATES
C       ROTATE COMPUTER MAN 90DEGREES ABOUT Z AXIS
        DO 2 K=1,6
        DO 2 J=1,55
        DO 2 I=1,110
        MAN2(I,J,K)=MAN(J,I,K)
2       CONTINUE
        CALL  DISCWT(FILE2,INVREC,1,MAN2)
        GO TO 11
C
C
C       PRINT OUT COMPUTER MAN IN MOSAIC FORM FROM NEW FILE
70      CONTINUE
        CALL DISCSU(3)
        DO 15 RECORD=1,14
        CALL  DISCRD(FILE2,RECORD,1,MAN2)
C       K IS EQUAL TO SLICE NUMBER WITHIN EACH RECORD
        DO 18 K=1,6
        DO 16 J=1,55
        DO 16 I=1,110
        WORK3(I,J)=MAN2(I,J,K)
16      CONTINUE
C       CALCULATE CROSS-SECTION NUMBER  (1-84)
        SLICE=((RECORD-1)*6)+K
        CALL MOSAIC(WORK3,SLICE)
18      CONTINUE
15      CONTINUE
        END
```


START
DO SLICE = 1, 6
READ IN FIRST CARD ARGUMENTS PERTAINING TO NEW SLICE CARDNUM, ARG1, ARG2, ARG3, ARG4, ARG5, ARG6
DO IROW = 1, ARG6
READ IN ADDITIONAL SLICE ARGUMENTS PERTAINING TO CURRENT ROW TISSUE CODES ARG7, ARG8, ARG9, ARRAY
DO I = 1, 24
IS ARRAY (N) IN THE RANGE (1-200)
N
ARRAY (N) = 0
Y
ARRAY(1). NE. 0
Y
N
DO N = 1, 23
ARRAY(N) = ARRAY(N+1)
PRINT OUT CONTENTS OF ARRAY, ARG7, ARG8, ARG9
CONSTRUCT SUBMATRIX FOR THIS SLICE
J = ARG7
NN = (ARG9-ARG8) + 1
DO I = 1, NN
IX = (ARG8+I) - 1
MAN (J, IX) = ARRAY(I)
PRINT OUT DATA TO CHECK ARRAY FOR ACCURACY
REFLECT SLICE ABOUT MIDPOINT OF X-AXIS
DO J = 1,55
DO I = 1,110
IX = 111-I
WORK1(J, I) = MAN(J, IX)
PRINT OUT REFLECTED DATA IN WORK ARRAY
B
2
COPY REFLECTED DATA ONTO MAIN MATRIX (MAN)
B
2
DO J = 1,55
DO I = 56,110
MAN(J, I) = WORK1(J, I)
REVERSE SLICE ORDER IN OUTPUT BUFFER (MAN1)
KM = 7-SLICE
DO J = 1,55
DO I = 1,110
MAN1(J, I, KM) = MAN(J, I)
ZERO WORK ARRAY AFTER EACH SLICE IS COMPLETED
DO IY = 1,55
DO I = 1,110
MAN(IY, I) = IZERO
WORK1(IY, I) = IZERO
INTERCHANGE (I, J) COORDINATES
DO K = 1,6
DO J = 1,55
DO I = 1,110
MAN3(I, J, K) = MAN1(J, I, K)
TRANSFER FOOT ADDITIONS TO NEW DISC PACK
RECORD = 1
OPEN DISC 55D22
CALL DISCDL(3, NAME2, DISC2)
WRITE FOOT SLICES ONTO NEW DISC FILE
CALL DISCWT(FILE2, RECORD, 1, MAN3)
PRINT OUT NEW ADDITIONS TO CHECK DISC FILE FOR ACCURACY
READ IN FOOT SLICES
CALL DISCRD(FILE2, RECORD, 1, MAN3)
DO K = 1,6
DO J = 1,55
DO I = 1,110
WORK3(I, J) = MAN3(I, J, K)
SLICE = ((RECORD -1)*6)+K
CALL MOSAIC(WORK3, SLICE)
STOP

```
      LIST(START)
$     MAXT(5)MINS
$     MAX3(200)LINES
$     COMM DISC 55D22 TO UNIT 3
      INTEGER CARDNUM,ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARRAY,WORK1,PLANE
      INTEGER ARG7,ARG8,ARG9
      INTEGER WDTH,HGHT
      INTEGER FILE2,DISC2
      INTEGER RECORD
      INTEGER SLICE
      INTEGER WORK3
      DIMENSION ARRAY(55),PLANE(50,20),WORK1(55,110)
      COMMON/WORK3/WORK3(110,55)
      COMMON/MAN3/MAN3(110,55,6)
      COMMON/MAN/MAN(55,110)
      COMMON/MAN1/MAN1(55,110,6)
      COMMON/MAN2/MAN2(55,110,6)
      COMMON/NAME2/NAME2(4)
      DATA IZERO/0/
      DATA DISC2/10H55D22 CMAN/
      DATA FILE2/10HINCAP BODY/
C     THIS SUBROUTINES READS THE ADDITIONAL FOOT
C     SLICES ONTO THE DISC FROM CARDS. THUS  THE
C     INCAPACITATION MODEL INCREASES IN HEIGHT
C     THIS PROGRAM ADDS 6 ADDITIONAL PORTIONS TO THE FEET
C     IN THE INCAPACITATION MAN
C     ARG1 = STARTING POSITION  OF SUBMATRIX  ON Y AXIS
C     ARG2 =       END POSITION  OF SUBMATRIX  ON Y AXIS
C     ARG3 = STARTING POSITION  OF SUBMATRIX  ON X AXIS
C     ARG4 =       END POSITION  OF SUBMATRIX  ON X AXIS
C     ARG5 = NUMBER OF COLUMNS FOR SUBMATRIX
C     ARG6 = NUMBER OF ROWS FOR SUBMATRIX
C     ARGUMENTS 7-9  PERTAIN TO EACH INDIVIDUAL ROW OF TISSUE CODES
C     ARG7 = Y COORDINATE OF ROW
C     ARG8 = STARTING X COORDINATE OF TISSUE CODES
C     ARG9 = X COORDINATE WHERE TISSUE CODES END
      DO 2000 SLICE=1,6
      READ(5,100)CARDNUM,ARG1,ARG2,ARG3,ARG4,ARG5,ARG6
100   FORMAT(I3,1X,6I4)
      DO 1000 IROW=1,ARG6
      READ(5,101) ARG7,ARG8,ARG9,(ARRAY(N),N=1,24)
101   FORMAT(I2,1X,I2,1X,I2,I2,23I3)
      DO 10 I=1,24
      IF((ARRAY(I).GE.1).AND.(ARRAY(I).LE.200)) GO TO 10
      ARRAY(I)=IZERO
10    CONTINUE
C     SHIFT TISSUE CODES UPWARDS IN INPUT BUFFER(ARRAY)
C     SUCH THAT ARRAY(1) CONTAINS AN ENTRY
C     CALCULATE INDEX TO RETRIEVE CORRECT ENTRY FROM ARRAY
C     CONSTRUCT OUR SUBMATRIX WORK1(Y,X) USING OUR ARGUMENTS
65    IF(ARRAY(1).NE.0)GOTO 75
      DO 60 N=1,23
      ARRAY(N)=ARRAY(N+1)
60    CONTINUE
      GOTO 65
75    CONTINUE
      WRITE(6,103) (ARRAY(I),I=1,24)
103   FORMAT(1 X,24I3)
```

```
C       TRANSFER CONTENTS OF ARRAY TO WORK1
C       USING ADDITIONAL DATA READ IN
C       INSERT ENTRY IN CORRECT LOCATION
C       CONSTRUCT SUBMATRIX FOR THIS SLICE
C       PRINT ARG7,ARG8,ARG9
        WRITE(6,104) ARG7,ARG8,ARG9
104     FORMAT(5X,3(I2,2X))
        WDTH=(ARG4-ARG3)+1
        HGHT=(ARG2-ARG1)+1
        J=ARG7
        NN=(ARG9-ARG8)+1
        DO 200 I=1,NN
        IX=(ARG8+I)-1
        MAN(J,IX)=ARRAY(I)
200     CONTINUE
1000    CONTINUE
C       PRINT OUT DATA TO CHECK ARRAYS FOR ACCURACY
        DO 300 K=1,55
        WRITE(6,102) (MAN(K,I),I=1,55)
300     CONTINUE
102     FORMAT(3 I3,/,25I3)
C       REFLECT SLICE ABOUT MIDPOINT OF X-AXIS THE LINE X=55
C       COPY REFLECTED DATA ONTO WORK MATRIX
        DO 500 J=1,55
        DO 500 I=1,110
        IX=111-I
        WORK1(J,I)=MAN(J,IX)
500     CONTINUE
C       PRINT OUT REFLECTED DATA IN WORK1 ARRAY
        DO 400 K=1,55
        WRITE(6,106)(WORK1(K,I),I=56,110)
106     FORMAT(3 I3,/,25I3)
400     CONTINUE
C       COPY REFLECTED DATA ONTO MAIN ARRAY (MAN)
        DO 600 J=1,55
        DO 600 I=56,110
        MAN(J,I)=WORK1(J,I)
600     CONTINUE
C       REVERSE SLICE ORDER IN OUTPUT BUFFER(MAN1)
C       FOOT SLICES WILL BE CROSS-SECTIONS 79-84
        KM=7-SLICE
        DO 900 J=1,55
        DO 900 I=1,110
        MAN1(J,I,KM)=MAN(J,I)
900     CONTINUE
C       ZERO WORK ARRAY AFTER EACH SLICE IS COMPLETED
        DO 105 IY=1,55
        DO 105 I=1,110
        MAN(IY,I)=IZERO
        WORK1(IY,I)=IZERO
105     CONTINUE
2000    CONTINUE
C       THESE ADDITIONAL FOOT SLICES AND THE PRECEEDING FOOT SLICES
C       ARE NOT CUTS MADE PARRALLEL TO X-Y PLANE
C
C       INTERCHANGE X-Y COORDINATES
C       ROTATE COMPUTER MAN 90DEGREES ABOUT Z AXIS
        DO 2 K=1,6
        DO 2 J=1,55
        DO 2 I=1,110
```

```
      MAN3(I,J,K)=MAN1(J,I,K)
2     CONTINUE
C     READ IN NEW DISC FILE 55D22 RECORD 1
      RECORD=1
      CALL DISCDL(3,NAME2,DISC2)
      CALL DISCWT(FILE2,RECORD,1,MAN3)
      CALL DISCRD(FILE2,RECORD,1,MAN3)
C     K IS EQUAL TO SLICE NUMBER WITHIN EACH RECORD
      DO 18 K=1,6
      DO 16 J=1,55
      DO 16 I=1,110
      WORK3(I,J)=MAN3(I,J,K)
16    CONTINUE
C     CALCULATE CROSS-SECTION NUMBER  (1-84)
      SLICE=((RECORD-1)*6)+K
      CALL MOSAIC(WORK3,SLICE)
18    CONTINUE
15    CONTINUE
      END
```


SUBROUTINE MOSAIC (KSI,I)
START
N=187
SEARCH NEW SLICE FOR TISSUE CODE ZERO
DO J = 1,55
DO L = 1,110
KSI(L,J) = 0
Y
GRID (L,J)=BLANK
A TISSUE CODE WAS ENCOUNTERED
N
MOD(L,10).EQ.6
N
Y
DO K = 1,N
IS TISSUE CODE STORED A VALID TISSUE CODE
MOD(J,10) = 0
N
Y
KSI(L,J) = CODE(K)
GRID(L,J) = PER
INVALID CODE
GRID(L,J) = ERROR
SUBSTITUTE SYMBOL FOR TISSUE CODE
GRID(L,J) = SYMBOL(K)
PRINT UPPER BORDER
PRINT [ORD(J), J=1,112]
DO L=1,55
PRINT SLICE
PRINT LEFT EDGE; GRID(L,J), J=1,110; RIGHT EDGE
PRINT LOWER BORDER
PRINT [ORD(J), J=1,112]
PRINT CROSS SECTION NUMBER(I)
RETURN

```
      SUBROUTINE MOSAIC(KSI,I)
      DIMENSION KSI(110,55),GRID(110,55),ORD(112)
      COMMON/SYMBOL/SYMBOL(200)
      COMMON/CODE/CODE(200)
      INTEGER KSI,GRID,SYMBOL,CODE,ERROR,BLANK,EDGE1,EDGE2,ZERO,
     1        PER,ORD
      DATA SYMBOL/
     A     1HQ,1HW,1HE,1HR,1HT,1HY,1HU,1HI,1HO,1HP,1HA,1HS,1HD,
     A     1HF,1HG,1HH,1HJ,1HK,1HL,1HZ,1HX,1HC,1HV,1HB,1HN,1HM,
     A     1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1HQ,1HW,1HE,
     A     1HR,1HT,1HY,1HU,1HI,1HO,1HP,1HA,1HS,1HD,1HF,1HG,1HH,
     A     1HJ,1HK,1HL,1HZ,1HX,1HC,1HV,1HB,1HN,1HM,1H0,1H1,1H2,
     A     1H3,1H4,1H5,1H6,1H7,1H8,1H9,1HQ,1HW,1HE,1HR,1HT,1HY,
     A     1HU,1HI,1HO,1HP,1HA,1HS,1HD,1HF,1HG,1HH,1HJ,1HK,1HL,
     A     1HZ,1HX,1HC,1HV,1HB,1HN,1HM,1H0,1H1,1H2,1H3,1H4,1H5,
     A     1H6,1H7,1H8,1H9,1HQ,1HW,1HE,1HR,1HT,1HY,1HU,1HI,1HO,
     A     1HP,1HA,1HS,1HD,1HF,1HG,1HH,1HJ,1HK,1HL,1HZ,1HX,1HC,
     A     1HV,1HB,1HN,1HM,1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,
     A     1H9,1HQ,1HW,1HE,1HR,1HT,1HY,1HU,1HI,1HO,1HP,1HA,1HS,
     A     1HD,1HF,1HG,1HH,1HJ,1HK,1HL,1HZ,1HX,1HC,1HV,1HB,1HN,
     A     1HM,1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1HQ,1HW,
     B     1H3,1HT,1HT,1HL,1HR,13*1H-/



      DATA CODE/
     A     2,3,4,5,6,23,7,8,9,10,24,11,12,
     B     13,14,179,15,16,17,18,173,174,19,20,21,22,
     C     25,55,26,27,28,29,30,31,32,175,176,33,34,
     D     35,36,37,38,39,40,41,42,177,178,43,44,45,
     E     46,47,48,49,50,51,52,53,54,73,55,56,57,
     F     58,59,60,61,62,63,64,65,66,67,68,75,69,
     G     70,71,72,73,96,74,76,77,78,79,180,80,81,
     H     82,83,84,85,86,87,88,89,99,100,90,91,92,
     I     93,94,95,96,119,97,98,101,200,102,103,104,105,
     J     106,110,111,107,108,109,112,113,114,115,116,117,118,
     K     119,120,121,122,123,124,125,126,127,128,129,130,131,
     L     132,133,134,135,136,137,138,139,140,141,142,146,143,
     M     144,145,147,149,150,151,152,153,154,155,156,157,158,
     M     159,160,161,162,163,164,165,166,167,168,169,170,171,
     N     172,181,182,183,184,13*0/
      DATA ERROR,BLANK,EDGE1,EDGE2,ZERO,PER/1H-,1H ,1H.,1H.,1HO,1H./
      DATA N /187/
      DATA ORD /112*1H./
```

```
      DO 20 J=1,55
      DO 20 L=1,110
               IF (KSI(L,J).EQ.0)GO TO 70
               DO 25 K=1,N
                  IF(KSI(L,J).EQ.CODE(K))GOTO 30
25             CONTINUE
               GRID(L,J)=ERROR
               GOTO 20
30             GRID(L,J)=SYMBOL(K)
20        CONTINUE
          WRITE(6,107)
107   FORMAT(' +Y')
          WRITE(6,106)(ORD(J),J=1,112)
106       FORMAT(112A1)
      DO 85 J=1,55
      JJ=56-J
      WRITE(6,106)EDGE1,(GRID(L,JJ),L=1,110),EDGE2
85        CONTINUE
          WRITE(6,106)(ORD(J),J=1,112)
          WRITE(6,100)I
100   FORMAT('0',46X,'CROSS SECTION',I5,35X,'+X')
      RETURN
70    GRID(L,J)=BLANK
      IF(MOD(J,10).EQ.6.AND.MOD(L,10).EQ.0)GRID(L,J)=PER
      GO TO 20
      END
```

DISTRIBUTION LIST

No. of Copies	Organization
12	Commander Defense Documentation Center ATTN: DDC-TCA Cameron Station Alexandria, VA 22314
1	Director of Defense Research and Engineering (OSD) Washington, DC 20305
1	Director Weapons Systems Evaluation Gp Washington, DC 20305
1	Chairman Defense Science Board Washington, DC 20301
1	Commander US Army Materiel Development and Readiness Command ATTN: DRCDMA-ST 5001 Eisenhower Avenue Alexandria, VA 22333
1	Commander US Army Materiel Development and Readiness Command ATTN: Mr. Cosgrove 5001 Eisenhower Avenue Alexandria, VA 22333
1	Commander US Army Aviation Research and Development Command ATTN: DRSAV-E 12th and Spruce Streets St. Louis, MO 63166
1	Director US Army Air Mobility Research and Development Laboratory Ames Research Center Moffett Field, CA 94035
1	Commander US Army Electronics Command ATTN: DRSEL-RD Fort Monmouth, NJ 07703
1	Director R&D Tech Support Activity ATTN: DRSEL-GG-DM Mr. C. Goldy US Army Electronics Command Fort Monmouth, NJ 07703
1	Commander US Army Missile Research and Development Command ATTN: DRDMI-R Redstone Arsenal, AL 35809
1	Commander US Army Tank Automotive Rsch and Development Command ATTN: DRDTA-RWL Warren, MI 48090
1	Commander US Army Mobility Equipment Research & Development Cmd ATTN: DRDME-WC Fort Belvoir, VA 22060
1	Commander US Army Armament Materiel Readiness Command ATTN: DRSAR-LEP-L, Tech Lib Rock Island, IL 61299
4	Commander US Army Armament Research and Development Command ATTN: DRDAR-LCS-TM, Mr. Gaydos DRDAR-SCH, Mr. H. Kahn DRDAR-SCN, Mr. Malinoski DRDAR-LCS, Mr. Einbinder Dover, NJ 07801

DISTRIBUTION LIST

No. of Copies	Organization
3	Commander US Army Armament Research and Development Command ATTN: DRDAR-TSS (2 cys) DRDAR-SCS, Mr. Mancini Dover, NJ 07801
1	Commander US Army Armament Research and Development Command ATTN: Ch, Large Caliber Lab DRDAR-LC Dover, NJ 07801
1	Commander US Army Armament Research and Development Command ATTN: Ch, Small Caliber Lab DRDAR-SC Dover, NJ 07801
1	Commander US Army Harry Diamond Labs ATTN: DRXDO-TI 2800 Powder Mill Road Adelphi, MD 20783
2	Commander US Army Materials and Mechanics Research Center ATTN: Dr. R. Frost Mr. E. DeLuca Watertown, MA 02172
2	Commander US Army Natick Research and Development Command ATTN: Mr. T. Keville Mr. M. Donomee Natick, MA 01760
1	Commander US Army Training & Doctrine Cmd ATTN: MAJ R. Young Fort Monroe, VA 23651
1	Office of Test Director Joint Services BO-EW-CW Test Program White Sands Missile Range NM 88002
1	Director US Army TRADOC Systems Analysis Activity ATTN: ATAA-SL, Tech Lib White Sands Missile Range NM 88002
1	Commander US Army Combat Developments Experimental Activity Fort Leavenworth, KS 66027
1	Commander US Army Armor School ATTN: Armor Agency Fort Knox, KY 40121
1	Commander US Army Aviation School ATTN: Aviation Agency Fort Rucker, AL 36362
1	Commander US Army Field Artillery School ATTN: Field Artillery Agency Fort Sill, OK 73405
1	Commander US Army Infantry School ATTN: ATSH-CS-MS-F, LTC Hurd Fort Benning, GA 31905
1	The Surgeon General Department of the Army Washington, DC 20310

DISTRIBUTION LIST

No. of Copies	Organization
1	HQDA (DAMO-CSS, MAJ Rogers) Washington, DC 20310
1	HQDA (DAMA-WSW, MAJ Woods) Washington, DC 20310
1	Chief of Naval Operations ATTN: OP-03EG Department of the Navy Washington, DC 20350
1	Commander US Naval Air Systems Command ATTN: AIR-604 Washington, DC 20360
2	Commander Naval Weapons Center ATTN: Code 753 Code 12 China Lake, CA 93555
1	Commander Naval Research Laboratory ATTN: Code 8434, Mr. Ferguson Washington, DC 20375
1	Commandant US Marine Corps Washington, DC 20380
1	Director, MCDEC ATTN: MAJ D. Crowe Quantico, VA 22134
1	AFATL (DLRW) Eglin AFB, FL 32542
1	Commandant US MCD (NATO) ATTN: LTC A.E. Bisantz US Member NATO Small Arms Tests Control Commission APO New York 09033
1	SRI International ATTN: Tim Thurgate, Rm J3041 333 Ravenswood Avenue Menlo Park, CA 94025

Aberdeen Proving Ground

Marine Corps Ln Ofc
Dir, USAMSAA
ATTN: Dr. J. Sperrazza
DRXSY-R, R. Simmons
DRXSY-GI, E. Heiss
DRXSY-M, R. Pollard
Cdr, USA CSL, EA
ATTN: Dr. Frank Shanty
Dr. Wm. Sacco
Cdr, USA TECOM
ATTN: C. Crider